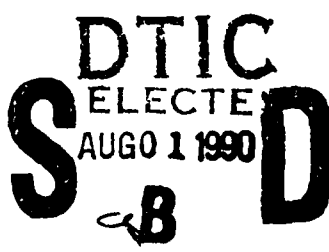


AD-A224 677

DTIC FILE COPY

①

REPORT DOCUMENTATION PAGE			Form Approved OMB No. 0704-0188	
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.				
1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE May 1990	3. REPORT TYPE AND DATES COVERED Thesis/Dissertation		
4. TITLE AND SUBTITLE An application of the A* Search to Trajectory Optimization		5. FUNDING NUMBERS		
6. AUTHOR(S) Craig K. Niiya				
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) AFIT Student at: Massachusetts Institute of Technology		8. PERFORMING ORGANIZATION REPORT NUMBER AFIT/CI/CIA - 90-033		
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) AFIT/CI Wright-Patterson AFB OH 45433		10. SPONSORING / MONITORING AGENCY REPORT NUMBER		
11. SUPPLEMENTARY NOTES				
12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for Public Release IAW AFR 190-1 Distribution Unlimited ERNEST A. HAYGOOD, 1st Lt, USAF Executive Officer, Civilian Institution Programs			12b. DISTRIBUTION CODE	
13. ABSTRACT (Maximum 200 words)				
				
14. SUBJECT TERMS			15. NUMBER OF PAGES 93	
			16. PRICE CODE	
17. SECURITY CLASSIFICATION OF REPORT UNCLASSIFIED	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT	

# An Application of the A\* Search to Trajectory Optimization

by

Craig K. Niiya  
Captain, United States Air Force

Submitted to the Department of Aeronautics and Astronautics  
on 11 May 1990 in partial fulfillment of the requirements for the Degree of  
Master of Science in Aeronautics and Astronautics

## ABSTRACT

As space operations become more complex and ambitious, there is a corresponding increase in the sophistication required of on-board algorithms for proximity operations. Unmanned missions such as planetary probes require sophisticated algorithms to deal with evolving mission requirements and contingencies, where man-in-the-loop control will be impractical. Other future missions that require autonomy for safety or security reasons will also require intelligent on-board controllers. This effort discusses a structure for a candidate proximity operations controller and provides initial development of an intelligent trajectory planner in three degrees of freedom using the A\* Node Search technique.

Thesis Supervisor (MIT): Professor Richard H. Battin, Adjunct Professor of Aeronautics  
and Astronautics

Thesis Supervisor(CSDL): Edward V. Bergmann, Flight Systems Section Chief, EGC



Announced	
Justification	
By	
Distribution/	
Availability Codes	
Dist	Avail and/or Special
A-1	

# An Application of the A\* Search to Trajectory Optimization

by

Craig K. Niiya  
Captain, United States Air Force

B.S., Mechanical Engineering  
Carnegie-Mellon University  
(1985)

Submitted to the  
Department of Aeronautics and Astronautics  
in Partial Fulfillment of the Requirements  
for the Degree of

Master of Science

at the

Massachusetts Institute of Technology

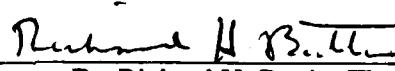
June 1990

© Craig K. Niiya, 1990. All Rights Reserved.

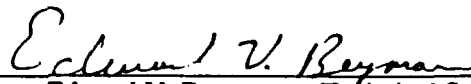
Signature of Author \_\_\_\_\_

  
Department of Aeronautics and Astronautics

Certified by \_\_\_\_\_

  
Dr. Richard H. Battin, Thesis Supervisor  
Adjunct Professor of Aeronautics and Astronautics

Certified by \_\_\_\_\_

  
Edward V. Bergmann, Technical Supervisor  
The Charles Stark Draper Laboratory, Inc.

Accepted by \_\_\_\_\_

Professor Harold Y. Wachman, Chairman  
Departmental Graduate Committee

## Acknowledgements

I am deeply grateful to Dr. Richard H. Battin for the interest which he took not only in my thesis, but in me. Indeed, it seems certain that his goal of sharing an awe and love for Astrodynamics has been accomplished in many of his students; I am privileged to have been one of them. I am also greatly appreciative of the many efforts extended on my behalf by Edward V. Bergmann, my Technical Supervisor at the Charles Stark Draper Laboratory. His insights and guidance have proved to be more than beneficial to any success I have had in pursuit of this thesis--his help has been essential. John Raquet, who is continuing with efforts to pursue this application of A\* technique further, has also volunteered his insights most graciously. More importantly, though, he is a good friend and a valued partner in the Gospel of Christ.

There are so many others at the Lab who have helped me with technical concepts, advice on presentation of my work, and who have, in short, been my friends, especially Todd Dierlam, Tony Bogner, Charles Cooke and Bruce Persson. As they say in Hawaii, Mahalo!--Thank you very much! I would also like to thank the lab for sponsoring this IR&D project and for providing this graduate fellowship which made my time here possible.

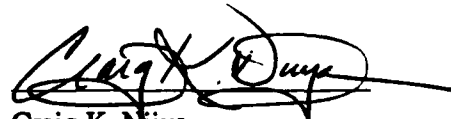
I also like to thank my many other friends that make up my church family, but in particular--David Johnson, Matthew Ide, David Lucia, Bruce Fottler and Melody Martin, Paul Christensen, David Gibb, Jeff Sanders, Jeff Bishop, and Lynn Thornton. Your love, support, and encouragement have been outstanding, but your willingness to share your lives and friendship with me, your ability to be vulnerable with me has endeared this time to me. I am forever grateful that you are truly my family (Mt. 12:50). There are so many who have contributed to my growth as a person, as an Air Force Officer, and most importantly as a friend of Christ's--to all of you, "thank you" seems hardly enough.

Each of us has reason to be grateful simply for our birth. Those of us that are fortunate enough to have received nurturing, understanding and support from our parents and families owe an even greater debt of love and gratitude. To my parents, my brothers and their wives and my two nieces, I love you all and I thank my Lord for you. I know with certainty few things, but I know this for sure: your investments in my life have been and continue to be foundational in all that I am and shall ever be. I am profoundly grateful.

Finally, and most importantly, I thank the LORD, my God. It is through Christ that I find reason to live and through Him that I find the energy and direction for the many pursuits in which He leads; it is through His mercy and grace that I am forgiven. You have given me love, hope, and faith. It is only fitting that I give you my life (Romans 12:1).

This report was prepared at The Charles Stark Draper Laboratory, under Draper Independent Research and Development (IR&D). Publication of this report does not constitute approval by the Charles Stark Draper Laboratory or the Massachusetts Institute of Technology of the findings or conclusions contained herein. It is published solely for the exchange and stimulation of ideas.

I hereby assign my copyright of this thesis to The Charles Stark Draper Laboratory, Cambridge, Massachusetts.

A handwritten signature in black ink, appearing to read 'Craig K. Niiya', with a long horizontal line extending to the right.

Craig K. Niiya  
Captain, US Air Force

Permission is hereby granted by The Charles Stark Draper Laboratory to the Massachusetts Institute of Technology to reproduce any or all of this thesis.

## **TABLE OF CONTENTS**

<b>1. <u>Introduction</u></b>	<b>5</b>
Proximity Operations	
Extending Mission Capability and Autonomy	
Intelligent Planner	
Optimal Control	
Hybrid Control Strategy	
Outline	
<b>2. <u>Background</u></b>	<b>13</b>
Transfer Problems	
Hill's Equations	
Derivation of Force-Free Hill's Equations	
The State Transition Matrix	
A*Search	
An Example of an A* Search	
<b>3. <u>A*Search: Trajectory Planning with Anomalies</u></b>	<b>26</b>
Grid Assignment Concepts	
Calculating the Cost	
Links to the Autopilot	
Implementation	
<b>4. <u>Testing</u></b>	<b>43</b>
The Simulation	
Testing Description	
Test Cases	
Case 1: Altitude Change Maneuver	
Case 2: V-Bar Maneuver	
Case 3: In-Plane Maneuver	
Case4: Out-of-Plane Maneuver	
Multiple Obstacles	
Comparison to a Two-Impulse Trajectory Solver	
Unplanned Disturbance	
<b>5. <u>Conclusions</u></b>	<b>64</b>
Summary	
Recommendations for Future Work	
<b><u>Appendix: Development of Solutions</u></b>	<b>71</b>
<b><u>References</u></b>	<b>87</b>

# **1 INTRODUCTION**

## **PROXIMITY OPERATIONS**

Space missions frequently require tasks such as rendezvous, docking, or stationkeeping, commonly referred to as proximity operations. By nature, these maneuvers are typically the most complicated and often the most fuel inefficient parts of the mission because of the high level of constraints placed on the motion of the vehicle.

Many current and future missions with vehicles such as the Space Transportation System (Space Shuttle) or the Orbital Maneuvering Vehicle (OMV) depend on safe and efficient proximity operations--the servicing or retrieval of satellites, or docking or stationkeeping with other spacecraft such as the space telescope or space station. Interplanetary missions such as the Mars Rover Sample and Return (MRSR) Mission will also depend on reliable proximity operations for mission success.

## **EXTENDING MISSION CAPABILITY AND AUTONOMY**

Spacecraft and space structures are rapidly increasing in complexity while the performance requirements continue to evolve to increasingly ambitious levels. Vehicles approaching a space structure may now be required to grapple a docking port located near "obstacles" such as solar arrays, antennae, or even inhabited appendages of the structure.

Servicing vehicles will increasingly be required to accomplish close rendezvous for retrieval of payloads for repair or return to Earth. Previous examples of such retrievals

## *Chapter 1. Introduction*

include the Solar Max repair on STS-41C and the Long Duration Exposure Facility (LDEF) recovery on STS-32.

The planning / replanning of the maneuvers which make up safe and reliable close proximity operations are therefore becoming more demanding activities, requiring the development of additional algorithmic tools to extend existing capabilities in this area. The technological gap between need and capability is especially visible in plans for truly autonomous vehicles where the problem-solving resources of man-in-the-loop control are unavailable. At present, the precision and flexibility required in proximity operations has been provided mainly by extensive training, simulation, and the resourcefulness of pilot and ground crews. However, unplanned events may force inherently inefficient trajectories to be followed in the absence of sophisticated replanning capabilities. In the case of jet failures, for example, the spacecraft commander might be constrained by procedure to go to a pre-planned, stable standoff point while earth based controllers would replan the operational trajectory. While in-flight replanning could have made use of spacecraft momentum prior to the failure, current operational policy might instead dictate that movement toward the target should be actively nulled, or perhaps, reversed to achieve the safe standoff condition.

Autonomy will become an increasingly critical capability when considering interplanetary missions such as the proposed Mars Rover Sample and Return (MRSR). In Martian orbit, the pilotless vehicle will have limited communication with its earth based controller due to the long signal delays between Earth and Mars. Earth-based replanning might be unavailable or impractical if an unplanned event (such as jet failure) occurred in Martian orbit; real-time command and control data will be unavailable since radio transmission times will be on the order of minutes.



## *Chapter 1. Introduction*

Implementing an in-flight replanning capability would require efficient use of computational assets. Real-time computations would be required to produce feasible and safe trajectories. Consequently, algorithms which were numerically simple would best accommodate these stringent real-time requirements. Planning algorithms which could also account for and possibly even take advantage of geometrical irregularities (such as effector coupling or spacecraft shape) could also contribute large dividends in fuel economy as well as time efficiency.

There is, then, a clear need to extend guidance and control systems capabilities for truly autonomous mission planning and implementation of safe and reliable control policies in real time.

## **INTELLIGENT PLANNER**

In the case of the Space Shuttle, very specific automatic maneuvers have been implemented for attitude control; translational maneuvers, however, are still executed manually. While a preliminary automated docking controller concept has been investigated, it is severely constrained by pre-specified initial conditions [1]. In European and Soviet efforts towards autonomous control, nominal cases are programmed with a level of robustness. The controller takes the form of a regulator, keeping the vehicle close to a nominal trajectory. If excessive variations arise in initial conditions, vehicle performance, or environmental constraints, that violate some pre-specified region of tolerance, the control system will typically resort to achieving a safe standoff location where replanning efforts can be directed by Earth-based controllers [2]. The degree of autonomy is clearly linked to the degree of fault accommodation and robustness to variations in mission, vehicle and environmental parameters and constraints.

## Chapter 1. Introduction

Unforeseen events such as malfunctions or off-nominal performance of a target vehicle could pose an extremely challenging docking task. The unpredictable motion of a disabled vehicle makes this type of "obstacle" a constraint that must be monitored and accommodated in real-time. While collision detection software does exist, that predicts location, time to impact, and type of collision [3], a unified approach to reconciling this data with other mission objectives and constraints has not previously been developed and implemented for on-board applications.

The planning architecture referred to in this thesis was envisioned to consolidate technology in automated maneuvers, collision prediction, and autopilot control systems to form an autonomous hierarchical planner [4]. The Autonomous Proximity Operations Planning System (APOPS), shown in Figure 1-1, uses three levels of decision responsibility with corresponding levels of mission authority. The Maneuver Manager (MM) makes executive (or strategic) decisions, the Trajectory Planner (TP) accomplishes policy (or tactical) decisions, and the Execution Manager (EM) implements policy within a specified cost horizon.

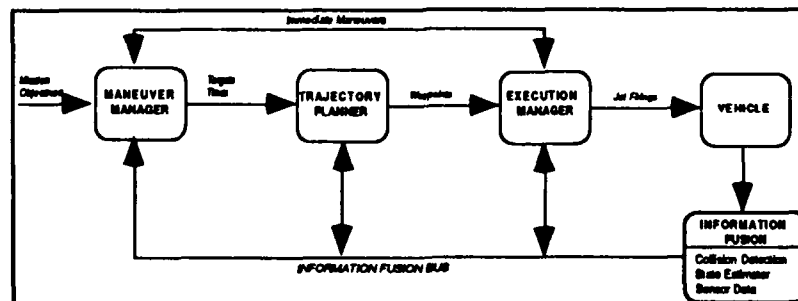


Figure 1-1. Three Tier Management

### MANEUVER MANAGER

The Maneuver Manager (MM) considers the strategic decisions associated with mission execution. In the face of changing requirements, vehicle status, or environment,

## *Chapter 1. Introduction*

the MM would choose whether to suspend execution of the current trajectory plan, to issue a work-around plan, or to stop and achieve a stable point to completely replan the operation. Based on a perceived advantage or need to replan a trajectory (e.g. an unforeseen obstruction to the current trajectory is detected by vehicle sensors), MM verifies that there is enough time to replan, and sends the Trajectory Planner a request for a new trajectory.

While the other two segments of this autonomous system would contain some algorithmic intelligence, MM would likely be the most dependent on programmed logic. A rule-based system could be an effective implementation of this logic since MM's decisions would be based on readily available and quantifiable information such as time to possible collision and jet performance / failure status. In response to this data, available through Information Fusion, MM logic would "optimize" the planner's performance by setting parameters such as grid meshsize or control effort granularity. Efficiency of computational effort may indeed be the difference between achieving the in-flight replanning capability and settling for the less efficient solution of moving to a stable standoff point and waiting for a new plan from the ground.

### **TRAJECTORY PLANNER**

When requested by the MM, the Trajectory Planner (TP) will execute a search for an optimal or near-optimal control policy which will safely take the spacecraft from its current trajectory to a desired target while accommodating mission, vehicle, and environment constraints. The TP makes decisions that may be considered "tactical" in the mission. Generating a waypoint string, an Optimal Search Algorithm (OSA) considers the orbital mechanics, the spacecraft dynamics, obstacle avoidance, vehicle control, and vehicle configuration status.

## *Chapter 1. Introduction*

There is a tacit assumption that, prior to prompting the Trajectory Planner, the MM is monitoring the execution of a policy which is optimal under current circumstances. Responding to a change in vehicle status, mission requirements, or environment, MM requests a replanning effort and TP initiates a search for a neighboring optimal control policy, accounting for new or changing constraints. Using fuel expenditure as the main component of the performance measure, the OSA compares various trajectory alternatives to arrive at a suitable control policy, in the form of a waypoint string. This string is then placed in an information buffer which is accessed by the next tier manager--the Execution Manager.

### EXECUTION MANAGER

The Execution Manager (EM) takes the trajectory solution generated by TP and generates control signals through a digital autopilot (DAP). The EM includes sensors and actuators as well as control logic. It is responsible for implementing the policies as directed by the higher managers. Along with data from the collision detector, Information Fusion receives information from the EM to generate vehicle and environment status reports as feedback to MM and TP.

### OPTIMAL CONTROL

The optimal control of any process requires a performance measure and will typically need to satisfy state and control constraints. For a given spacecraft design, payload or maneuvering capacity is mainly determined by fuel considerations. Thus, the performance measure should serve to minimize fuel expenditures. Also, while transfer times are not fixed, it may be desirable to minimize the maneuver interval as well. Constraints on state and control variables are mainly generated by safety concerns (such as vehicular collisions and plume impingement) and vehicle capabilities (i.e. jet authority).

## **HYBRID CONTROL STRATEGY**

While traditional optimal control techniques can develop off-line policies which are optimal or near optimal, typically the computation required of such algorithms will render them impractical for autonomous proximity operations where in-flight mission planning is desired. In real-time, where faster algorithms are needed, heuristic node search techniques employing simple numerical operations can be used to obtain neighboring near-optimal control. Further, by exploiting parallel processing technology, the efficiency of these algorithms could be significantly enhanced.

While a node search technique could independently derive a refined optimal or near optimal control, it would then require an extensive node space, yielding again, an impractical algorithm. A more efficient planner concept would use a traditional optimal control technique to develop a nominally optimal trajectory off-line while in-situ replanning would be accomplished by a node search technique to develop neighboring near-optimal solutions in the presence of evolving mission uncertainties and constraints. It is toward this strategy that we have developed the A\* trajectory planner concept .

## **OUTLINE**

This effort to implement an intelligent planner concept incorporates a linearized model of orbital motion called the Clohessy-Wiltshire Equations and a node search technique called A\*. The planner discussed in this thesis starts with a transfer time which represents a nominally optimal two-burn trajectory. It pieces together a multi-segment trajectory which avoids obstacles and accounts for vehicular limitations by using penalty functions on node costs.

## *Chapter 1. Introduction*

Previous work has been done to implement the hierarchical planner concept and the A\* search logic in the area of tactical aircraft mission planning [5,6]. Building from that experience, this thesis seeks to implement the A\* search algorithm in the close proximity operations area using an innovative node generation scheme to accommodate vehicle capabilities and evolving physical constraints.

## **2 BACKGROUND**

### **TRANSFER PROBLEMS**

Past research has investigated restricted transfer problems in Orbital Mechanics. Two problems which have received considerable study are the Hohmann Transfer and Lambert's Problem. The Hohmann Transfer is a fuel optimal transfer in altitude which does not allow arbitrary specification of phasing constraints or transfer time. The Hohmann Transfer moves a satellite from a circular orbit to a coplanar circular orbit at a different altitude as the vehicle moves through half an orbit [7]. Because the Hohmann Transfer is a solution to a very specific problem, in the past, orbital maneuvers might have been planned in one of two ways. Solving the altitude change first, the phasing adjustment would be accomplished with a V-bar maneuver, providing a very stable approach to the target. A second approach might be to target an altitude above or below the target to accomplish phasing. If the target was behind of the chase vehicle, a higher orbit would be targeted and orbital mechanics would cause the chase vehicle to fall back relative to the target. The opposite would be done if the target was ahead of the chase vehicle.

Lambert's problem has also received considerable attention. It deals with the transfer of a vehicle from one orbit to another with phasing and time specified. There are an infinity of trajectories between two points on different orbits, varying only by transfer time (and correspondingly, orbital energy). However, a transfer is uniquely specified given the locations of the two endpoints and the transfer time [8].

This trajectory planner seeks a unique solution to the less constrained problem which calls for an optimal transfer between specified initial and terminal states while transfer time is left unspecified.

## HILL'S (CLOHESSY-WILTSHIRE) EQUATIONS

The Euler-Hill equations describe motion in the three translational degrees of freedom for proximity operations. The force-free solution of these equations have become known as the Clohessy-Wiltshire equations. The coordinate system used in this thesis is commonly referred to as Local Vertical, Local Horizontal or LVLH reference frame; it is a simple rotation of the Hill's coordinate system. (See Figure 2-1.)

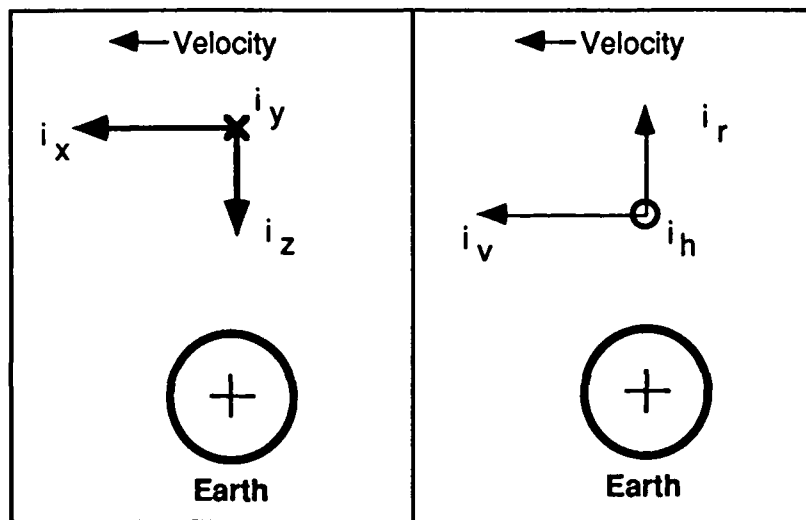


Figure 2-1. LVLH, Hill's Coordinate Systems

Two of the three translational components  $i_x$ , the unit vector along the velocity vector, and  $i_z$ , the unit vector pointing to the center of attraction, are defined within the orbital plane while the third,  $i_y$ , is the unit vector opposite the orbit angular momentum vector.

## DERIVATION OF FORCE-FREE HILL'S EQUATIONS [9]

Newton's second law for a body's motion defined in a inertial frame of reference states that the acceleration of that body will be determined by the sum of the forces on the body.



## Chapter 2. Background

$$\ddot{\mathbf{r}} = \frac{1}{M} \Sigma \text{ Forces on the Satellite} \quad \langle 2.1 \rangle$$

We will assume that the jets used to achieve velocity changes are impulsive and that these equations of motion will be used to describe motion between these impulsive burns.

Thus equation 2.1 reduces to:

$$\ddot{\mathbf{r}} = -\frac{GM_{\text{Earth}}}{r^3} \mathbf{r} \quad \langle 2.2 \rangle$$

Making the change of coordinates to the LVLH frame we observe the following definitions:

- 1) let  $\rho$  be the position vector of a point in the LVLH frame:

$$\rho = x\mathbf{i} + y\mathbf{j} + z\mathbf{k}$$

- 2) let  $\omega$  be the orbit angular momentum vector in the LVLH frame:

$$\omega = -\omega\mathbf{j}$$

NOTE:  $\omega$  is a constant since we assume a circular orbit.

We have the relative velocity in the LVLH frame given by:

$$\mathbf{v}_{\text{LVLH}} = \frac{d\rho}{dt} + (\omega \times \rho) \quad \langle 2.3 \rangle$$

where the time derivative represents only the time rate of change in position within the reference frame and does not account for the motion of the coordinate system.

Expansion of equation 2.3 results in:

$$\mathbf{v}_{\text{LVLH}} = (\dot{x} + \omega z)\mathbf{i} + \dot{y}\mathbf{j} + (\dot{z} - \omega x)\mathbf{k}$$

## Chapter 2. Background

The inertial acceleration is given by:

$$\mathbf{a}_{\text{INERTIAL}} = \frac{d\mathbf{v}_{\text{LVLH}}}{dt} + (\boldsymbol{\omega} \times \mathbf{v}_{\text{LVLH}}) + \mathbf{a}_{\text{LVLH}} \quad \langle 2.4 \rangle$$

where the acceleration,  $\mathbf{a}_{\text{LVLH}}$ , is the centripetal acceleration of a body in circular motion, given by  $\omega^2 \mathbf{R}$  ( $\mathbf{R}$  = nominal orbital radius for the origin of the LVLH frame).

$$\mathbf{a}_{\text{INERTIAL}} = (\ddot{x} + 2\omega\dot{z} - \omega^2 x)\mathbf{i} + \ddot{y}\mathbf{j} + (\ddot{z} + 2\omega\dot{x} - \omega^2 z + \omega^2 R)\mathbf{k} \quad \langle 2.5 \rangle$$

Expressing equation 2.2 into LVLH coordinates we find:

$$\mathbf{\ddot{r}} = \frac{GM_{\text{EARTH}}}{(\sqrt{x^2 + y^2 + (z-R)^2})^3} (x\mathbf{i} + y\mathbf{j} + (z-R)\mathbf{k})$$

We now make some simplifying assumptions to linearize these expressions.

Noting that  $\rho \ll R$ , the magnitude of the acceleration components in the  $\mathbf{i}$  and  $\mathbf{j}$  directions can be well approximated by:

$$\ddot{r}_x \approx \frac{GM_{\text{EARTH}}}{R^3} x \approx -\omega^2 x \quad \langle 2.6a \rangle$$

and,

$$\ddot{r}_y \approx \frac{GM_{\text{EARTH}}}{R^3} y \approx -\omega^2 y \quad \langle 2.6b \rangle$$

However, in the  $\mathbf{k}$  direction this approximation is not sufficiently accurate. To find a useful approximation, we use the binomial series to expand the expression. The expansion formula is:

$$[1+k]^m = 1 + km + o[k^2]$$

## Chapter 2. Background

Applying this to the expression,

$$\begin{aligned}\ddot{r}_z &= -GM_{\text{EARTH}} (z-R) (x^2 + y^2 + (z-R)^2)^{-3/2} \\ &= -\frac{GM_{\text{EARTH}}}{R^3} (z-R) \left( 1 - 2\frac{z}{R} + \frac{x^2 + y^2 + z^2}{R^2} \right)^{-3/2} \\ &= -\frac{GM_{\text{EARTH}}}{R^3} (z-R) \left[ 1 + \frac{3}{2} \left( \frac{2z}{R} - \frac{x^2 + y^2 + z^2}{R^2} \right) \right]\end{aligned}$$

We have again invoked the assumption that  $\rho \ll R$  and dropped terms of order  $(x^2 + y^2 + z^2)/R^2$  and higher. After expanding, we drop the non-linear term  $(z^2/R)$ , using the same rationale, to arrive at:

$$\ddot{r}_z \approx \frac{GM_{\text{EARTH}}}{R^3} (2z+R) \approx \omega^2(2z+R) \quad <2.7>$$

We now equate the expressions for the components in equation 2.6a, 2.6b and 2.7, with the components in equation 2.5 and obtain:

$$\begin{aligned}\ddot{x} &= 2\omega\dot{z} \\ \ddot{y} &= -\omega^2 y \\ \ddot{z} &= -2\omega\dot{x} + 3\omega^2 z\end{aligned}$$

The linearized equations of motion expressed in matrix/vector form are:

$$\dot{\mathbf{v}} = \omega \begin{bmatrix} 0 & 0 & 2 \\ 0 & 0 & 0 \\ -2 & 0 & 0 \end{bmatrix} \mathbf{v} + \omega^2 \begin{bmatrix} 0 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 3 \end{bmatrix} \mathbf{r}$$

The assumptions used in deriving these equations are:

## Chapter 2. Background

- 1) The displacement of the point of interest from the origin of the LVLH frame is small compared to the radial distance of the origin from the attractive force,  $\rho \ll R$ . With  $\rho$  on the order of  $10^3$  feet and  $R$  on the order of  $10^7$  feet, this is a safe assumption.
- 2) There is negligible force on the object. Since the jet firings are on the order of 10 seconds or less and the period between firings is 500 seconds or more, the velocity changes can be regarded as impulsive.
- 3) The orbital rate is approximately constant. In a circular orbit,  $\omega$  is a constant. Some of the maneuvers require introduction of a slight eccentricity, however the orbits are still nearly circular and this assumption is still valid.

### THE STATE TRANSITION MATRIX

The state transition matrix relates a state vector at a certain point in time with a state vector at some specified later time. For the three translational directions in the LVLH frame, the state vector is a six-vector including the displacement and the velocity of the point relative to the reference frame.

$$\mathbf{x}(t) = \Phi(t, t_0) \mathbf{x}(t_0)$$

The matrix,  $\Phi(t, t_0)$ , is a solution of the linear differential equation on the state vector [8]:

$$\frac{d\mathbf{x}}{dt} = \mathbf{F}(t) \mathbf{x}$$

$$\frac{d}{dt} \Phi(t, t_0) = \mathbf{F}(t) \Phi(t, t_0)$$

## Chapter 2. Background

The state transition matrix for Hill's Equations,  $\Phi(t, t_0)$ , indexed from  $t_0=0$  is [10]:

$$\begin{bmatrix} 1 & 0 & 6\omega t - 6\sin\omega t & \frac{4\sin\omega t}{\omega} - 3t & 0 & \frac{2(1-\cos\omega t)}{\omega} \\ 0 & \cos\omega t & 0 & 0 & \frac{\sin\omega t}{\omega} & 0 \\ 0 & 0 & 4-3\cos\omega t & \frac{2(\cos\omega t-1)}{\omega} & 0 & \frac{\sin\omega t}{\omega} \\ 0 & 0 & 6\omega(1-\cos\omega t) & 4\cos\omega t-3 & 0 & 2\sin\omega t \\ 0 & -\omega\sin\omega t & 0 & 0 & \cos\omega t & 0 \\ 0 & 0 & 3\omega\sin\omega t & -2\sin\omega t & 0 & \cos\omega t \end{bmatrix}$$

## A\*SEARCH

The A\* search, developed by Hart, Nilson, and Raphael in 1968, is a modified tree search. Trees typically represent a family history where information characterizing previous generations has a direct effect on present and future generations. In the orbital trajectory optimization context, generations represent the physical states which the spacecraft can achieve at a particular time. As the search progresses, the object is to proceed along a family line which will be the optimal path to the goal state.

Many algorithms have been developed to accomplish tree searches. One of these is the A\* search, which uses information about the problem scenario to form heuristic estimates which directs the search along the most promising directions. While the simpler depth first or breadth first search algorithms would be exhaustive (i.e. they would look at all possible family lines and therefore arrive at the globally optimal solution), A\* also guarantees an optimal solution over the given node space while implementing a comparatively more efficient search technique. Similarly, the Dynamic Programming

## Chapter 2. Background

technique while well researched would, in general, be a less efficient methodology for use as part of an intelligent planner.

A depth first search would be an effective technique where there are many options at each node, but relatively few nodes in the solution string (see Figure 2-2). It would not, however, be as desirable in a search space with long solution strings and relatively few options at each node (see Figure 2-3). (The reverse situations apply to the breadth first search.)

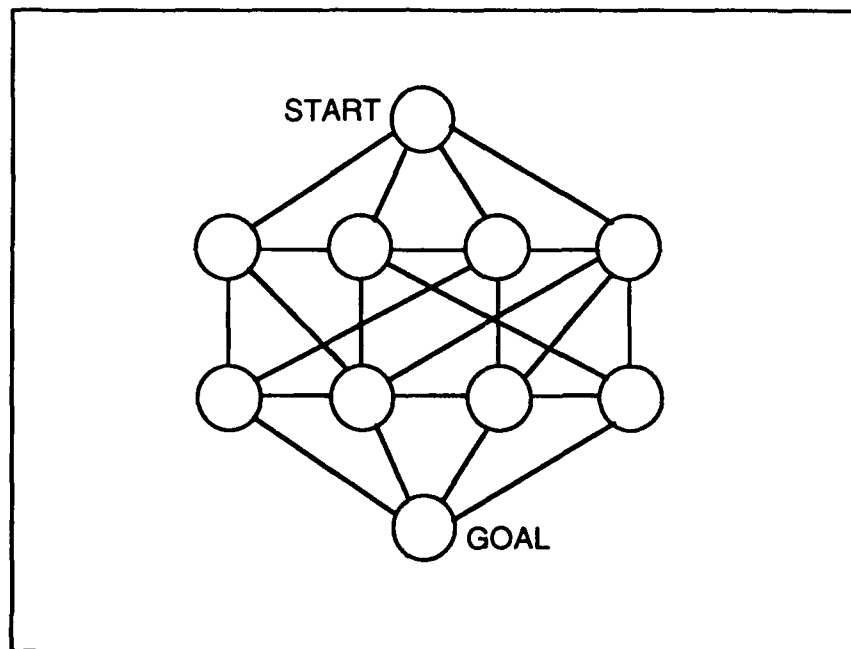


Figure 2-2. Node space with short solution string

By comparison, A\* avoids wasting time searching unproductive areas of node space by placing emphasis on the most promising directions. This effect is accomplished by using a heuristic cost estimate to identify the best node candidates. The search begins as the start node is *expanded*. In expanding a node, the possible successor nodes are identified (*generated*) and the actual costs of getting to those nodes are computed. Along

## Chapter 2. Background

with the actual cost of going from the parent to the child node, A\* adds a best guess of the cost to complete a solution from that child node by using information about the vehicle, the environment, and the particular mission.

A\* then rank orders the generated (successor) nodes on a memory stack called the OPEN List and places the expanded (parent) node on a separate stack called the CLOSED List. The search is continued by pulling the next node to be expanded from the top of OPEN. The search is terminated when the goal node is pulled from OPEN. The solution string is then generated by backtracking from child to parent, starting with the goal and ending at the start node.

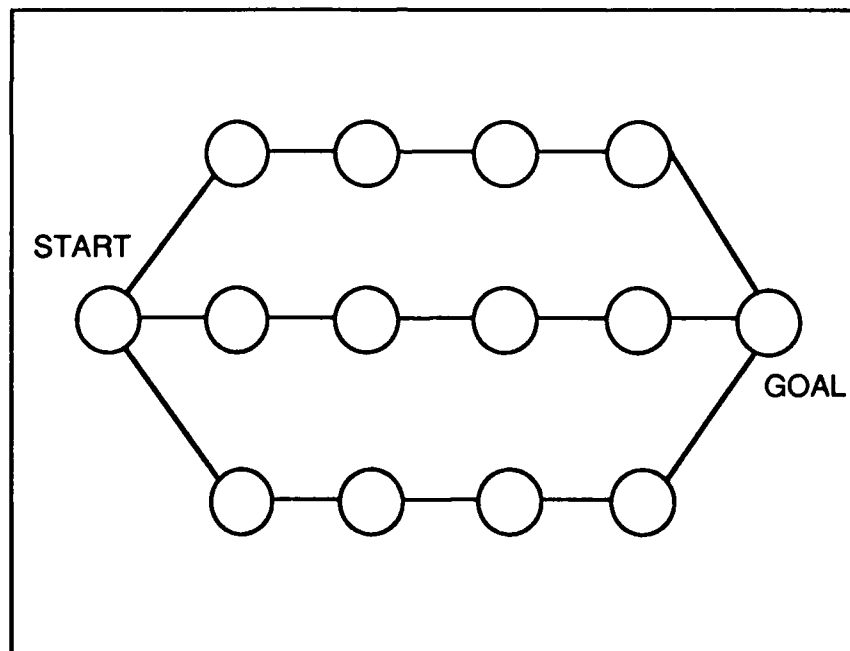


Figure 2-3. Node space with long solution string

During the search, if there are multiple routes to a given node, then as paths (strings) are generated, the information set for a particular node reflects the most efficient

## *Chapter 2. Background*

path to get there. A\* has been proven guaranteed optimal over the search space if the heuristic is *admissible* [11]. A heuristic cost is admissible if the estimate is less than or equal to the actual cost. *Aggressive* heuristics, which are close to the actual cost, accelerate the search by providing better information about the cost to successfully accomplish the specified maneuver. Such estimates, however, are usually accompanied by higher computational cost.

The A\* methodology is demonstrated in a sample problem formulation--the Roadmap Problem, where the object is to get from a start node to a goal node using a network of roads. The heuristic estimate is the air distances between the nodes.

### **AN EXAMPLE OF AN A\* SEARCH: THE ROADMAP PROBLEM**

Consider the history of the roads in the Boston area; most of the roads are asphalt versions of what used to be cow trails connecting the various towns in a more or less tangled network. A frequent challenge for the automobile commuter is to find the quickest route from one town to another, which may involve accounting for the presence of an obstacle.

The start node is Wayland. The goal is to get to Lexington before the shot is heard 'round the world. (See Figure 2-4a.) The start node is the first parent node and is the first to be *expanded*. In expanding a node, its successors are generated. This means that "children" nodes have information tags associated with them that identify their parent node (PATH), the actual cost to go from the parent to the child node (GCOST), and an estimate as to the cost to complete the journey to the goal from the child node (HCOST). In this problem formulation, GCOST is given by:  $g(N,N') = \text{actual distance by road between the}$



## Chapter 2. Background

two nodes. (N represents the parent node and N' a child node.) The heuristic estimate, H<sub>COST</sub>, is given by  $h(N') =$  the line-of-sight distance (or "*as the crow flies* ") from the the child node to the Goal.

The search begins by putting the start node (Wayland) on the OPEN stack. The OPEN stack is a rank-ordered list of candidate nodes for expansion. These nodes are ordered by their total cost, F<sub>COST</sub>:  $f(N,N')=g(N,N')+h(N')$ . The start node is the first node to be expanded; in this problem, the generation function simply tells A\* to consider the towns geographically adjacent to Wayland: Lincoln, Concord, and Waltham (see Figure 2-4b). A\* now places the parent node (Wayland) on the CLOSED stack (the listing of nodes that have been expanded) and generates the children nodes. After computing the actual distances and estimating the air distance between each of the town centers, A\* ranks Lincoln, with a total cost (F<sub>COST</sub>) of 18, as number one for expansion on the OPEN stack.

While the colonists think that they have found the optimal path to Lexington, unforeseen to them is the sabotage of a bridge by the repressors of religious freedom. Because the bridge is unpassable, it represents an obstacle. As Lincoln is expanded and put on the CLOSED list, the generation of Lexington as a child of Lincoln is penalized because of the presence of this obstacle (see Figure 2-4c). Consequently, while the F<sub>COST</sub> for Lexington as a child node of Lincoln would have been 19, it is now 49 because A\* has assigned a obstacle penalty of 30 to this route. A\* continues with the expansion of Lincoln as Concord and Waltham are generated. Since the costs of these nodes as children of Lincoln exceed the generations from Wayland, these nodes are removed (or *pruned* ) from the OPEN list. Similarly, if a node on CLOSED was found to be an inferior generation it would be removed from the CLOSED list, assigned the new generation information (PATH, G<sub>COST</sub>, H<sub>COST</sub>, and F<sub>COST</sub>) and entered again on the OPEN list. At this point, number one on the OPEN list is Concord.

## *Chapter 2. Background*

The search continues as Concord is expanded (see Figure 2-4d). The children nodes for Concord are Bedford, Lincoln, and Lexington. As previously noted, A\* avoids extra effort by pruning duplicate states from both the OPEN and CLOSED lists. In this generation, the new child node, Lincoln from Concord is compared to an earlier node generation (Lincoln from Wayland) and is found inferior. The Lincoln from Concord node is therefore never entered on the OPEN list. After completing the expansion of Concord, A\* observes that the FCOST for Lexington as a child node of Concord (as opposed to being a child node of Lincoln) now places Lexington on the top of OPEN. A\* then terminates the search as the Goal has been pulled from the top of the OPEN stack.

Observing the final configuration of the planning road map in Figure 2-4d, we note that two possible nodes were not expanded, Bedford and Waltham. Were an exhaustive node search or a Dynamic Programming method used, the computational costs of expanding these nodes would have been required, unless a degree of optimality was to be sacrificed.



### **3 A\* SEARCH: TRAJECTORY PLANNING WITH ANOMALIES**

The implementation of an optimal search algorithm using the A\* search posed two major challenges. The first was the creation of an efficient and meaningful node space; the second was a method of determining the cost of getting from one node to another node. This chapter also characterizes the link between the planner and autopilot, ending with a summary of the implementation of the various concepts and tools introduced in this investigation of the A\* technique.

#### **GRID ASSIGNMENT CONCEPTS**

In developing an efficient and meaningful grid, two approaches were investigated reflecting different relative emphasis on two design considerations--the representation of the full range of options available and the preclusion of nodes that were physically unachievable or redundant. These roughly correspond to the Linear Algebra concepts of the spanning and linear independence of a space, the ideal gridspace corresponding to a basis.

Spatial Grid Assignment: The first concept generated a uniform grid of nodes. A regular lattice structure was constructed to represent a linear discretization of the space surrounding the line of sight between the start and goal nodes. A variation to the concept included nodes "behind" the start node (i.e. in directions opposite from the goal) and "beyond" the goal node to provide options for longer transfers or delays before requiring a jet firing. As a further improvement, node density was varied to provide a finer distribution of choices in the vicinity of the start, goal, or obstacle locations. In these areas, small changes in trajectory could result in large changes in cost. It therefore made sense to increase the node concentration in these areas.

The motion required by the linear grid (see Figure 3-1) did not, in general, correspond to unforced trajectories. Many velocity changes were made so that the vehicle would pass through the points represented by this uniform grid, rather than follow the natural trajectories defined by orbital mechanics. The end effect was a trajectory which required trim burns at each node to account for the discontinuities associated with the linear discretization. To reduce the numerous jet cyclings, the grid was altered to surround a nominally optimal two-impulse trajectory, creating a node space with more curvature.

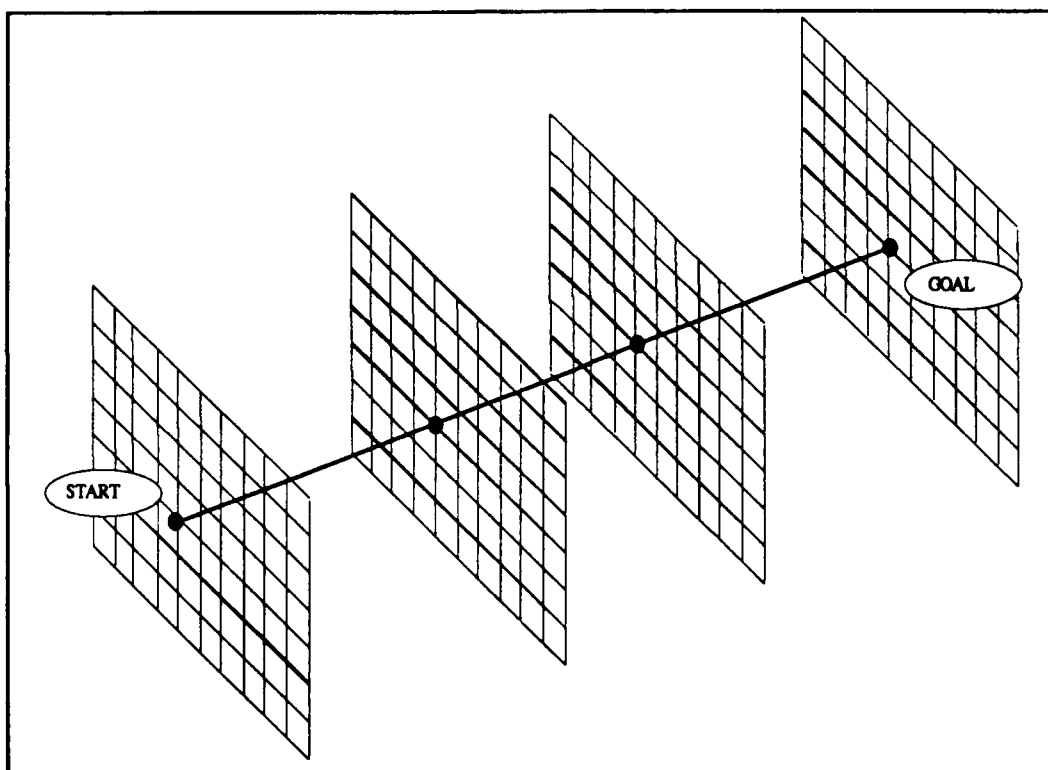


Figure 3-1. Spatial Grid Assignment--a linear discretization of space.

These efforts significantly enhanced A\* performance. Even with these improvements, however, the search occasionally developed trajectories that caused the vehicle to enter an infinite looping trajectory (between three or four nodes), appearing to

### Chapter 3. A\*Search

mimic cycloid motion. These trajectories were caused in part by the discretized nature of the spatial grid. Rather than producing a travelling-circle solution characterized by the cycloid, the planner, confined to the static spatial grid produced a closed loop trajectory. While cycloid motion specified a node somewhere in between two nodes which the spatial grid provided, the planner was constrained to choose a node which did not corresponded to a coasting trajectory. If the node which turned out to be the lower cost alternative happened to complete a closed curve, an infinite looping trajectory was created, producing an unuseable solution.

Spatial Grid assignment created a node space which represented the full range of spatial locations available to the planner, at the expense of a severe computational burden. Significant developments using pruning techniques would have been necessary to make the Spatial Grid Assignment concept workable. Further difficulties were encountered when considering the discretization of time. Each node could lie on any of infinitely many actual trajectory arcs, each differing in energy and transfer time. Picking the transfer time from this infinity of possibilities became a substantial task. While we could have picked a suitably small range of times to limit the task, there was no guarantee that the optimal time would be in this range.

Dynamic Grid Assignment: A more efficient method of creating a meaningful grid space uses vehicle dynamics and orbital mechanics to generate candidate nodes in the vehicle's state space (i.e. the six-space reflecting the three translational displacements and velocities) as opposed to the Spatial Grid Assignment approach which represents displacements only. Starting with the initial conditions on the start node ( $x_0, y_0$  specified), a nominal trajectory is computed using a nominally optimal transfer time (NOMTIM) for a two-impulse solution.

### Chapter 3. A\*Search

The nominal trajectory determines a velocity required,  $\underline{v}_R$ , at the node being expanded for a vehicle to arrive at the goal location in the specified transfer time. A jet select algorithm [12], which models vehicle and actuator dynamics, then computes the quantized firing times and records the specified jets. The quantized firing times are stored in a 3-vector called the quanta vector,  $\underline{q}$ . Each component of the quanta vector is varied, using a 27 point stencil, by a pre-specified percentage (e.g. 20%) so that the individual components are given 20% greater, 20% less, or identical value to the nominal to create the next generation of nodes (see Figure 3-2). Up to twenty-six distinct variations on the nominal quanta vector,  $\underline{q}(N')$  are generated by applying permutations of a weighting vector, for example:

$$\underline{q}(N'') = \underline{q}(N') + [0 \ 0.2 \ 0] \begin{bmatrix} \underline{q}(N')_1 & 0 & 0 \\ 0 & \underline{q}(N')_2 & 0 \\ 0 & 0 & \underline{q}(N')_3 \end{bmatrix}$$

$$\underline{q}(N''') = \underline{q}(N') + [0.2 \ 0 \ 0] \begin{bmatrix} \underline{q}(N')_1 & 0 & 0 \\ 0 & \underline{q}(N')_2 & 0 \\ 0 & 0 & \underline{q}(N')_3 \end{bmatrix}$$

The minimum variation amount in any component is one quanta. Because these vectors represent quantized jet firing intervals, if any component of  $\underline{q}(N')$  is negative because of the variations, that vector is discarded.

In addition to the nominal trajectory and the twenty-six variations, a twenty-eighth option is added corresponding to no firing:  $\underline{q}(N^{(28)}) = \underline{0}$ . This node allows the vehicle to simply coast during consecutive time increments, giving the planner the option of creating no-fire periods where the chase vehicle waits for a more opportune time to execute a burn. As an example, consider the case where the chase and the target vehicles are not in the same orbit plane. It may be beneficial to wait until the chase vehicle is in the plane to execute a burn that reduces the remainder of the rendezvous problem to an in-plane maneuver. Each

### Chapter 3. A\*Search

of these twenty-eight options represents a different trajectory available to the planner as a follow-on segment from the parent node in building a multi-burn solution.

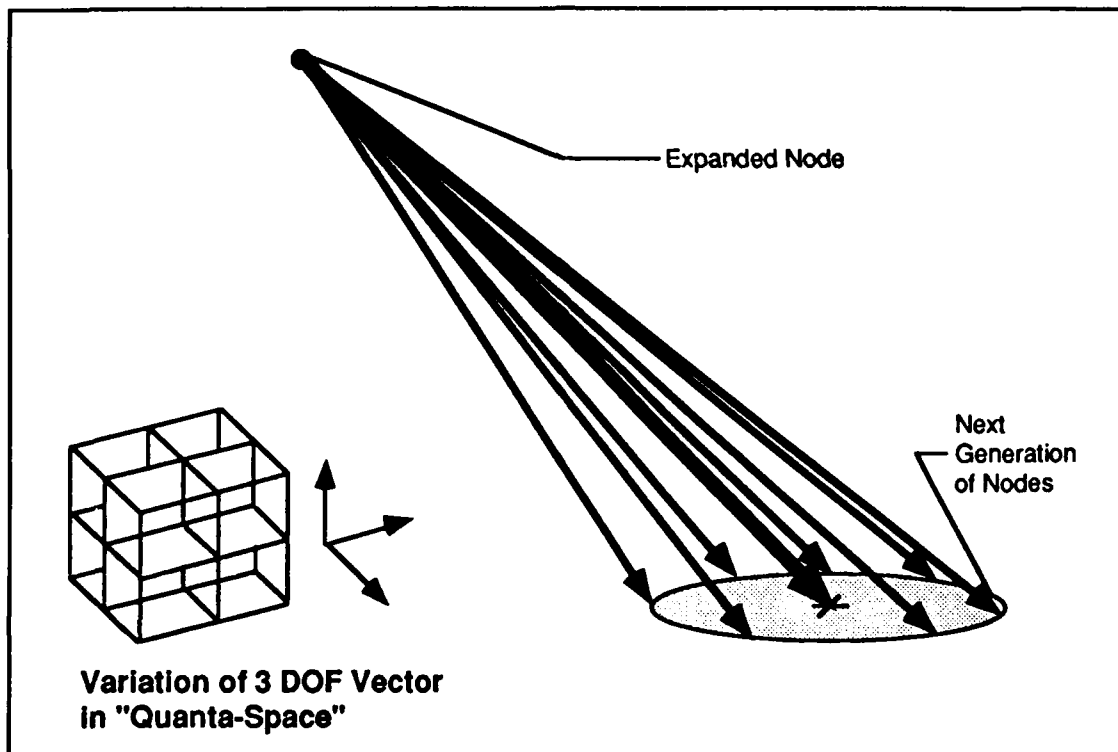


Figure 3-2. Generation of variations on the nominal trajectory.

To generate these trajectories, the specified impulsive velocity change is applied to the parent node's state vector. Using the state transition matrix, the node is then propagated forward in time by a set time increment (DELTAT) which is a fraction of the nominal transfer time (NOMTIM) to generate a child node. The new state is then recorded. In total, each node that is expanded generates up to 28 successors including the 26 variations on the nominal trajectory, the no firing option and the nominal trajectory (which will put the vehicle on a coasting trajectory to the goal).

The method is labelled "dynamic" for two reasons. First, it reflects the dynamics of the vehicle. As such, the node pattern or grid space is unique to the vehicle. The



### *Chapter 3. A\*Search*

significance is that the vehicle can actually achieve the states associated with every node generated. Second, nodes are generated as the search proceeds. This directly implies that the grid is an evolving structure and is unique, not only to the vehicle, but to the problem as well. By contrast to the spatial grid concept, where the node pattern is predetermined and independent of the problem scenario, the dynamic grid is flexible and responsive to the specific geometries and constraints of the given situation.

The node pattern using dynamic grid assignment is inherently more efficient as it generates nodes that are located in the natural trajectories given by orbital mechanics. By involving the vehicle and actuator dynamics in node generation we implicitly account for effects such as jet discretization and can easily accommodate vehicle configuration changes such as failed jets by updating the jet selection data base. Further, since the grid only includes *physically reachable* states (in physical and velocity space), we have essentially pruned the large number of unreachable nodes encountered in the spatial grid assignment.

Although a dense grid that represents all possible control combinations offered by the vehicle could conceivably have been created, on something with as many redundant effectors as the space shuttle, far too many nodes would have been generated. If, for example, at each node the full range of jet combinations (forty-four effectors taken three at a time) was generated, rather than twenty-eight successors we would have thirteen thousand. Realizing that some of these possible successors would not produce useful trajectories, it is clear that a more manageable subset of the full range of combinations (such as the quanta variation scheme with twenty-eight successors), while not exhaustive, would be more desirable.

While optimizing the fuel consumption during the planning of the mission is an important objective, it should not be the overriding focus of a planner. The requirements

### *Chapter 3. A\*Search*

for safety and assurance of mission success should take precedence over local optimization of fuel consumption. The goal of this design effort was to arrive at a neighboring optimal controller which could safely avoid significant obstacles that evolve with time such as the malfunctioning satellite which may have erratic movement. In order to attain this capability, the search must be computationally efficient. In pursuit of the real-time capabilities desired, we favor the dynamic grid assignment concept and as sparse a grid as possible. The obvious counter concern is that sufficient variations are provided to the search for it to develop an perturbed trajectory that will allow successful circumnavigation of any obstacles encountered. The resolution of these competing concerns should be reflected in the specification of parameters set by the Maneuver Manager's program logic.

## **CALCULATING THE COST**

The second major development, the method of calculating the cost of node propagation, is a very natural outgrowth of using the jet select algorithm in node generation.

The A\* search algorithm depends on the calculation of two costs, a generation cost,  $GCOST(N,N')$ , and an heuristic cost,  $HCOST(N')$ . In both calculations the primary function is the generation of the velocity required,  $\underline{v}_R$ , which is determined by the two node locations and a transfer time. In generating  $GCOST(N,N')$ , the velocity required at the parent node to arrive at the child node in the specified time increment ( $\Delta T$ ) is calculated and the velocity change between the coasting velocity at the parent node,  $\underline{v}_C$ , and the velocity required,  $\underline{v}_R$ , is passed to the jet select algorithm, which calculates the quanta vector. The quanta required for the velocity change is stored as the generation cost of the child node,  $N'$ , from the parent node,  $N$ :  $GCOST(N,N') = QUANTA_G ( (\underline{v}_R \text{ at } N) - (\underline{v}_C \text{ at } N) )$ .

### Chapter 3. A\*Search

To compute the heuristic cost,  $H\text{COST}(N')$ , a similar operation is done where the child node,  $N'$ , takes the place of the parent node and the goal node takes the place of the child node. In addition to the cost of the burn at  $N'$ , we also add the cost of the deceleration burn to null the coasting velocity at the goal node as well.  $H\text{COST}(N') = \text{QUANTA}_H((\underline{V}_R \text{ at } N') - (\underline{V}_C \text{ at } N')) + \text{QUANTA}_H((\underline{V}_C \text{ at } \text{GOAL}))$ . The transfer time is the difference between the nominal time ( $\text{NOMTIM}$ ) and the flight time at  $N'$ . If  $N'$  is along a lengthy search string the flight time may be equal to or greater than  $\text{NOMTIM}$ . If this is the case the transfer time to the goal is one  $\text{DELTAT}$ .

It is important to note that in calculating the heuristic there is an assumption made regarding the admissibility of the estimate. The admissibility requirement is that the heuristic cost must be less than or equal to the actual cost of completion. A more aggressive heuristic would seek to provide a very close estimate of the actual cost. In this development, the formulation of the  $H\text{COST}$  quanta function is not the same as the one used in the  $G\text{COST}$  calculation. In order to speed up the computation, we use an ideal jet implementation rather than invoking the jet select algorithm for the two velocity changes. The effect is to make the heuristic less aggressive. Biasing the estimate in the opposite direction, the possible non-optimality of an immediate two-burn trajectory between the child node and the goal may produce a heuristic cost that is larger than the actual cost. This calls into question the admissibility of the estimate. The heuristic calculates the cost for a two-burn transfer in a specific transfer time. While the transfer time is almost certainly non-optimal, the two-burn trajectory is itself non-optimal in some cases. The effects of the inadmissibility of the heuristic have been noted in some test cases, indeed resulting in the planning of a non-optimal trajectory.

While an admissible heuristic could be easily produced by simply scaling the existing heuristic, this would decrease the aggressiveness of the heuristic. Alternatively, an

exact and optimal solution from each candidate node to the goal could be calculated by a traditional method such as variational calculus, but this would be time consuming and defeat the intent of a real time planner, creation of an efficient algorithm. Instead, noting that the more significant objectives of mission success and vehicle safety are attained by this optimizing planner, the aggressiveness and computational efficiency of using a two-burn trajectory as the heuristic is chosen over possible admissibility with a different estimator.

## LINKS TO THE AUTOPILOT

As a part of the APOPS system, the Trajectory Planner sends a waypoint file to a trajectory buffer for the Execution Manager to access. This waypoint file contains three items of information: the waypoint in the form of the chase vehicle's state vector, the time tag associated with the waypoint, and a flag that states whether the waypoint is a forced fire point. As opposed to a reference waypoint which is provided solely for the error regulation, the forced fire point corresponds to a node contained in the solution string which requires a velocity change.

The Digital Autopilot (DAP) reads the waypoint file using it as a reference path. The DAP functions as a regulator, using the waypoints and the time tags to compare the actual state vector to the planned (reference) trajectory, keeping the vehicle state vector within a pre-specified error bound. While the jet select used in the search algorithm is the same as used in the DAP, the planned firings are adjusted or augmented to regulate error. Finite quantization of jet impulses, imperfect modelling and disturbances prevent the autopilot from precisely nulling position and rate errors. In order to cause the vehicle to move from one end of the error sphere to another (so that average position is close to the reference), a small perturbation in the requested jet firing is implemented. This causes the

vehicle to limit cycle along the reference trajectory as it moves toward the target. In order that planner-requested firings (which may be on the same order of magnitude as these limit cycle firings) are not ignored, a flag is provided to force a jet firing at those specific waypoints corresponding to the solution nodes in the search's grid space.

## IMPLEMENTATION

As discussed in chapter 2, the dynamics model used in this effort is the linearized Clohessy-Wiltshire equations of motion. This dynamics model accounts for accelerations on bodies due only to the gravitational attraction of a spherical earth. The state transition matrix form of the solution to the Clohessy-Wiltshire equations is used to generate candidate nodes that represent the states in the three translational degrees of freedom and the three associated translational velocities. The search employs the developments discussed earlier in this chapter-- a grid that contains these nodes and a cost function and heuristic estimate on which the A\* search algorithm is based. The inputs to the search are (1) the initial conditions on the vehicle and any obstacles, (2) the terminal conditions on arrival at the goal, (3) a nominal transfer time, (4) the variation factor (which governs the percentage of variation of the quanta vector), and (5) a time constraint on the trajectory.

### DETERMINING THE NOMINAL TRAJECTORY

The search starts by computing a nominal trajectory from the start node to the goal node using the two-impulse trajectory associated with the nominal time. The velocity change required at the start node is calculated so that the vehicle's state vector will reflect the velocity required,  $\mathbf{v}_R$ , to achieve the nominal trajectory to the target. The calculation of  $\mathbf{v}_R$  proceeds from the  $\Phi$  matrix, evaluated at the nominal time. If we divide the matrix into

### Chapter 3. A\*Search

quadrants we can rearrange the equations to compute the velocity needed to pass through a desired node.

$$\begin{bmatrix} x_1 \\ y_1 \\ z_1 \\ \dot{x}_1 \\ \dot{y}_1 \\ \dot{z}_1 \end{bmatrix} = \begin{bmatrix} \phi_{11} & \phi_{12} \\ \phi_{21} & \phi_{22} \end{bmatrix} \begin{bmatrix} x_0 \\ y_0 \\ z_0 \\ \dot{x}_0 \\ \dot{y}_0 \\ \dot{z}_0 \end{bmatrix}$$

$$\underline{v}_R(\underline{x}_0, \underline{x}_1) = \begin{bmatrix} \dot{x}_0 \\ \dot{y}_0 \\ \dot{z}_0 \end{bmatrix} = \phi_{12}^{-1} \left( \begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix} - \phi_{11} \begin{bmatrix} x_0 \\ y_0 \\ z_0 \end{bmatrix} \right)$$

$\underline{v}_R(\underline{x}_0, \underline{x}_1)$ , then represents the velocity required at the start node  $(x_0, y_0, z_0)$  so that a spacecraft will arrive at the goal state  $(x_1, y_1, z_1)$  in the nominal time.

The nominal velocity change required is the difference between the required velocity,  $\underline{v}_R(\underline{x}_0, \underline{x}_1)$ , and the current coasting velocity vector,  $\underline{v}_C$ . (In the expansion of the start node,  $\underline{v}_C = \underline{v}_0$ . In expanding subsequent parent nodes,  $\underline{v}_C$  corresponds to the velocity at that node, resulting from the previous coasting trajectory.) This velocity change request is then passed to a linear programming algorithm for jet selection, which in this case is identical to the DAP's jet select algorithm. (The linear programming problem is to solve for  $\underline{x}$ , where  $\underline{w} = A\underline{x}$ , such that  $x_i \geq 0$  for all  $x_i$ , and  $\sum f_i x_i$  is minimized.) The output of the jet select is a vector of on-times for specified jets.

### VARYING THE QUANTA AND GENERATING THE CHILD NODES

As described earlier in this chapter, this vector of quantized firing times, called the quanta vector, is altered by the variation factor (the percentage called for at the initiation of

### Chapter 3. A\*Search

the program) using the 27 point stencil and the no-firing option. The twenty-eight on-time combinations,  $q(N^i)$  are then converted into velocity changes. If  $A$  is an acceleration matrix and  $q(N^i)$  is the quanta vector which generates the  $i$ th node, the expression for the new parent velocity vector is:

$$v_p(N^i) = v_c + Aq(N^i).$$

A new state vector at the parent node is defined by augmenting the displacement components of the parent node,  $I_p=[x_p, y_p, z_p]$  with the new velocity vector,  $v_p(N^i)=[\dot{x}_p(N^i), \dot{y}_p(N^i), \dot{z}_p(N^i)]$ . The new parent vector is propagated using the state transition matrix through one time interval (DELTAT) to arrive at the child node,  $x(N^i)$ . (Note:  $[x_p, y_p, z_p]=[x_0, y_0, z_0]$  in the start node expansion.)

$$\begin{bmatrix} x(N^i) \\ y(N^i) \\ z(N^i) \\ \dot{x}(N^i) \\ \dot{y}(N^i) \\ \dot{z}(N^i) \end{bmatrix} = [\Phi(t+\Delta t, t)] \begin{bmatrix} x_p \\ y_p \\ z_p \\ \dot{x}_p(N^i) \\ \dot{y}_p(N^i) \\ \dot{z}_p(N^i) \end{bmatrix}$$

This process of creating a parent velocity vector,  $v_p(N^i)$ , and augmenting the displacement vector,  $I_p=[x_p, y_p, z_p]$ , with it to form a new parent vector, occurs for each child node generation. This illustrates the differing natures of the two grid assignment schemes discussed earlier. Nodes created using Dynamic Grid Assignment (the method presented) are actually generated by implementable control effort and lie on natural, unforced trajectories.

## COMPUTING GCOST AND HCOST

There are two components to the total cost, FCOST, which is the index used in sorting the candidate nodes. The generation cost, GCOST, is the quanta for the desired velocity change which puts the chase vehicle on a coasting trajectory from the parent to the child node:  $GCOST(N,N') = QUANTA_G [\underline{v}_P - \underline{v}_C]$ , where  $\underline{v}_P$  is the parent velocity vector and  $\underline{v}_C$  is the coasting velocity at the parent node.

The heuristic estimate, HCOST, represents the ideal firing times associated with the two impulsive velocity changes required to put the chase vehicle on a coasting trajectory from the child node to the goal and terminate the maneuver at that point. Given the child node's state vector,  $\underline{x}(N^i)$ , computed in the last section, the algorithm then computes the velocity required at the child node:

$$\underline{v}_R(\underline{x}(N^i), GOAL) = \Phi_{12}^{-1} \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} - \Phi_{11} \begin{bmatrix} \underline{x}(N^i) \\ \underline{y}(N^i) \\ \underline{z}(N^i) \end{bmatrix},$$

where the  $\Phi$  sub-matrices are computed with the appropriate terminal time. In this development, the terminal time is the nominal time (one of the inputs to the planner), unless the time index for the child node is already greater than or equal to the nominal time. If this is the case, the terminal time is extended by one DELTAT beyond the current time. As in the GCOST calculation, the impulsive velocity change required is computed to achieve the specified  $\underline{v}_R$ .

The second impulsive velocity change is required to null the chase vehicle's projected terminal velocity at the end of the heuristic coasting trajectory. By propagating the child node's altered state vector (i.e. the child node's displacement vector augmented



### Chapter 3. A\*Search

with the velocity required,  $\underline{v}_R(\underline{x}(N^i), \text{GOAL})$  forward in time, we get the projected terminal state and the chase vehicle's terminal velocity.

$$\begin{bmatrix} x_F(N^i) \\ y_F(N^i) \\ z_F(N^i) \\ \dot{x}_F(N^i) \\ \dot{y}_F(N^i) \\ \dot{z}_F(N^i) \end{bmatrix} = \begin{bmatrix} \phi_{11} & \phi_{12} \\ \phi_{21} & \phi_{22} \end{bmatrix} \begin{bmatrix} x(N^i) \\ y(N^i) \\ z(N^i) \\ \dot{x}_R(N^i, \text{GOAL}) \\ \dot{y}_R(N^i, \text{GOAL}) \\ \dot{z}_R(N^i, \text{GOAL}) \end{bmatrix}$$

and,

$$\underline{v}_F = (\dot{x}_F(N^i), \dot{y}_F(N^i), \dot{z}_F(N^i)).$$

Taking the sum of these two impulsive changes, and dividing by the specific thrust of a single jet, we get an idealized jet firing time.

$$\text{HCOST} = (|\underline{v}(N^i) - \underline{v}_R(\underline{x}(N^i), \text{GOAL})| + |\underline{v}_F - \underline{v}|) / \text{specific thrust}$$

HCOST represents the ideal jet firings where the vehicle's effectors are located exactly along the direction of the velocity change required and have no minimum on-times. (Typically, jets will have a minimum firing time; firing requests which fall below that threshold are not implemented, or implemented by an effector with a different geometry. The Space Shuttle's threshold of 80 milliseconds is the basis for the quantized burn interval defined earlier in the chapter.)

### CHECKING FOR OBSTACLE INTERCEPTION

To check for obstacles that may intercept the trajectory of the chase vehicle, we take an obstacle's state vector and propagate it in time up to the current flight time.

$$\Delta_{\text{OBST}}(t) = \Phi(t, t_0) \Delta_{\text{OBST}}(t_0)$$

During the time interval between the parent and child nodes the state vectors of the chase vehicle and the obstacle(s) are propagated in increments of one twentieth of the time increment, DELTAT. The obstacles have been modelled as uniform spheres, so if the magnitude of the difference between the displacement components of the state vectors is smaller than the sum of the chase vehicle and obstacle radii, a collision has been predicted and a penalty is applied to  $\text{GCOST}(N, N^i)$ . The effect of this penalty is to increase the cost of this node and which causes this candidate to be sorted toward the bottom of the OPEN list. As a possible alternative, future implementations of the algorithm could simply remove the node from the list altogether. In practice, a collision is unacceptable, so carrying this node at the bottom of the OPEN list merely adds to the computational baggage of the search.

#### SORTING THE NODE AND STORING REQUIRED INFORMATION

Summing the generation and heuristic costs (including any penalty from a collision with an obstacle), we get the total cost,  $\text{FCOST} = \text{GCOST} + \text{HCOST}$ . The child node is now sorted on the OPEN list where the node reflecting the least costly alternative at the top of the stack.

Additional information is also stored in reference to the newly generated node. The child node's parent is recorded on  $\text{PATH}(N^i)$ , for use at the termination of the search to recover the solution string. The current flight time is computed by adding the time increment, DELTAT, to the parent's flight time and the result is stored in  $\text{FLTTIM}(N^i)$ . After each of the variations,  $q(N^i)$ , have been used to generate the child nodes, the expansion of the parent node is complete and the parent is placed on CLOSED.

### Chapter 3. A\*Search

The search continues as the next parent node is pulled off the top of the OPEN stack. Before expanding the next node, the associated flight time is checked against the time constraint, TIMCON. If after adding one DELTAT to the flight time, the time constraint is violated, the parent node is discarded and the search continues with the next node on the OPEN list. (A time constraint is a typical feature of fuel optimal control problems; if time is unconstrained or not penalized, solution trajectories often produce a do nothing strategy since it is the lowest cost alternative [13].)

#### TERMINATING THE SEARCH

The A\* search algorithm typically terminates when the goal node is pulled from the OPEN stack. However, because control effort discretization and computational inaccuracies contribute to produce inherent numerical errors, this application of the search algorithm terminates the search when the candidate parent node is physically located within a sphere that surrounds the goal. The goal sphere in this implementation has a radius of 3 feet.

Following termination of the search, the solution string is generated. Using the PATH list, which contains pointer identifying the parents of each node, the algorithm backtracks beginning with the last node. The solution string is completed when the start node is recovered.

#### GENERATING THE WAYPOINT FILE

The final task for the planner is the generation of a waypoint file. Beginning with the start node, the solution states,  $\mathbf{x}_{ss}$ , are propagated in time steps of 60 seconds to generate a reference trajectory for the DAP,  $\mathbf{x}_{wp}$ :

### Chapter 3. A\*Search

$$\begin{aligned}\mathbf{x}_{wp}(t+60) &= \Phi(t+60, t_0) \mathbf{x}_{ss}(t_0), \\ \mathbf{x}_{wp}(t+120) &= \Phi(t+120, t_0) \mathbf{x}_{ss}(t_0), \dots\end{aligned}$$

The propagation from the start node continues until the time at which the next state in the solution string is reached. At that point, the next node on the solution string takes the place of the start node and the base time reference for the  $\Phi$  matrix is the time index (FLTTIM) associated with that node.

For each of the nodes in the solution string,  $\mathbf{x}_{ss}$ , a flag is set to identify that waypoint as a forced firing point requiring the DAP to implement the best velocity change possible to rectify the vehicle's actual state vector with the reference trajectory. The last waypoint entry is a forced firing to achieve the goal state; the chase vehicle arrives at the origin and nulls its closing velocity.

## **4 TESTING**

### **THE SIMULATION**

Four test cases were simulated on the existing Draper Space Systems Simulator to demonstrate the Trajectory Planner's capability to successfully maneuver to an orbiting target while avoiding obstacles. The Space Systems Simulator is a high fidelity simulation of on-orbit vehicular motion. The gravity model used for all of the test cases was a spherical earth. To demonstrate disturbance accommodation, the J2 gravity term, due to an oblate earth, was used to introduce a disturbance which had not been accounted for in the trajectory planner's dynamics model. The simulator can also account for the environmental effect of gravity gradient torque on vehicle attitude, but does not account for atmospheric effects such as aerodynamic drag. The differential equations of motion in the six degrees of freedom are independently integrated by a fourth-order Runge-Kutta algorithm.

The vehicles used in these simulations are the Space Shuttle (the maneuvering or chase vehicle), the Orbital Maneuvering Vehicle (the obstacle or intercepting vehicle), and the Hubble Space Telescope (the target vehicle). While the attitude dynamics and orientation can be arbitrary in the simulation, shuttle attitude was held fixed, aligned with the LVLH coordinate frame so that the nose of the shuttle was pointed along the velocity vector.

### **TESTING DESCRIPTION**

In each of the four test cases, the planner started with a nominal transfer time, from which it produced a nominal execution plan. To test the ability of the planner to revise plans to accomplish obstacle avoidance, an intercepting trajectory was computed for the

## *Chapter 4. Testing*

plans to accomplish obstacle avoidance, an intercepting trajectory was computed for the obstacle which would place it directly in the chase vehicle's nominal path. Given the obstacle's trajectory, A\* then developed a perturbed trajectory for the pursuer which avoided the obstacle. (See the Appendix for a sample planning history where the planner makes use of information about the obstacle.)

In the problem statement, the initial and terminal states may be assigned arbitrary displacements and velocities. In these cases, the initial states were given velocities corresponding to nearly circular Keplerian orbits and the target state was located at the origin of the LVLH coordinate system. The planner's task is to specify a trajectory for the chase vehicle from the initial offset position to the target, while avoiding any obstacles, meeting a constraint on terminal velocity, and optimizing the solution generated from the node space. Data is presented reflecting the fuel expenditures associated with the nominal transfers as well as the perturbed trajectories which accomplish obstacle avoidance. Trace plots show the path taken by the chase vehicle during the transfers. A representation of the obstacle is shown in both the nominal and obstacle runs to demonstrate the collision of the obstacle with a vehicle on a nominal trajectory and the successful avoidance of the intercepting obstacle on the perturbed trajectory.

A multiple obstacle problem with three intercepting vehicles is also posed to demonstrate the flexibility of the A\* Trajectory planner. An additional comparison with the solution of a simple two-impulse trajectory solver demonstrates the effectiveness of the optimization accomplished by A\*. Finally, disturbance accommodation, readily available to the integrated APOPS system, is illustrated by introducing an unplanned disturbance to the waypoint execution on the simulator.

## TEST CASES

**Case 1:** Altitude Change Maneuver-- (0,0,-3000) to the origin. The chase vehicle is 3000 feet above the target vehicle along the positive radius vector.

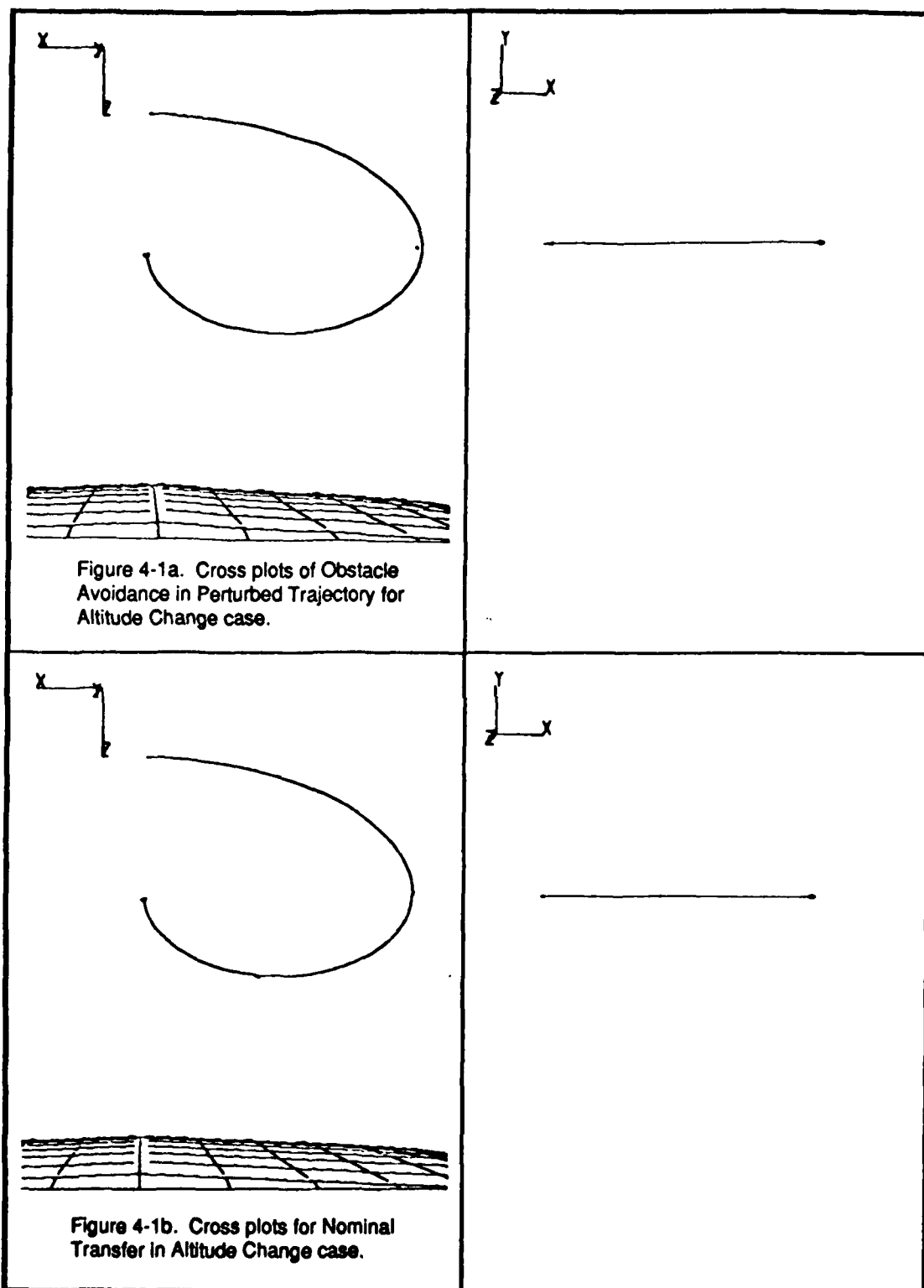
After running the algorithm with various transfer times and selecting the best result, the nominal time was set at 3750 seconds. The problem described in this case is a departure from the Hohmann Transfer which is accomplished in half an orbit (approximately 2700 sec). While the Hohmann maneuver accomplishes a fuel optimal altitude change, it does not allow specification of phasing requirements; although it would have nulled the displacement in the radial (z) direction, the Hohmann transfer would not have caused the vehicle to arrive at the origin at the specified time. (The vehicle would have arrived at the same altitude and in the same orbit plane, but either leading or trailing the target.) The performance of the A\* trajectory planner in this case is compared to a direct solution of the Clohessy-Wiltshire equations later in this chapter.

Case	Initial Position* [ft]	Initial Velocity [ft/sec]	Nominal Time [sec]
1	0	-5.22322	3750
	0	0	
	-3000	0	
	Fuel Use [lbs.] (No Obstacle)	Fuel Use [lbs.] (Obstacle)	Execution Time [sec] No Obst/Obst
	169.1	174.2	3750/3750

Table 4-1. Summary of Case 1

\*All displacements and velocities are expressed in the LVLH coordinate system (Downrange, Out-of-Plane, and Vertical)

The trace plots in Figure 4-1 clearly demonstrate the obstacle avoidance capability provided by the A\* planner. The nominal trajectory passes directly through the obstacle, while the perturbed trajectory puts the chase vehicle on a path which is a safe distance away





#### Chapter 4. Testing

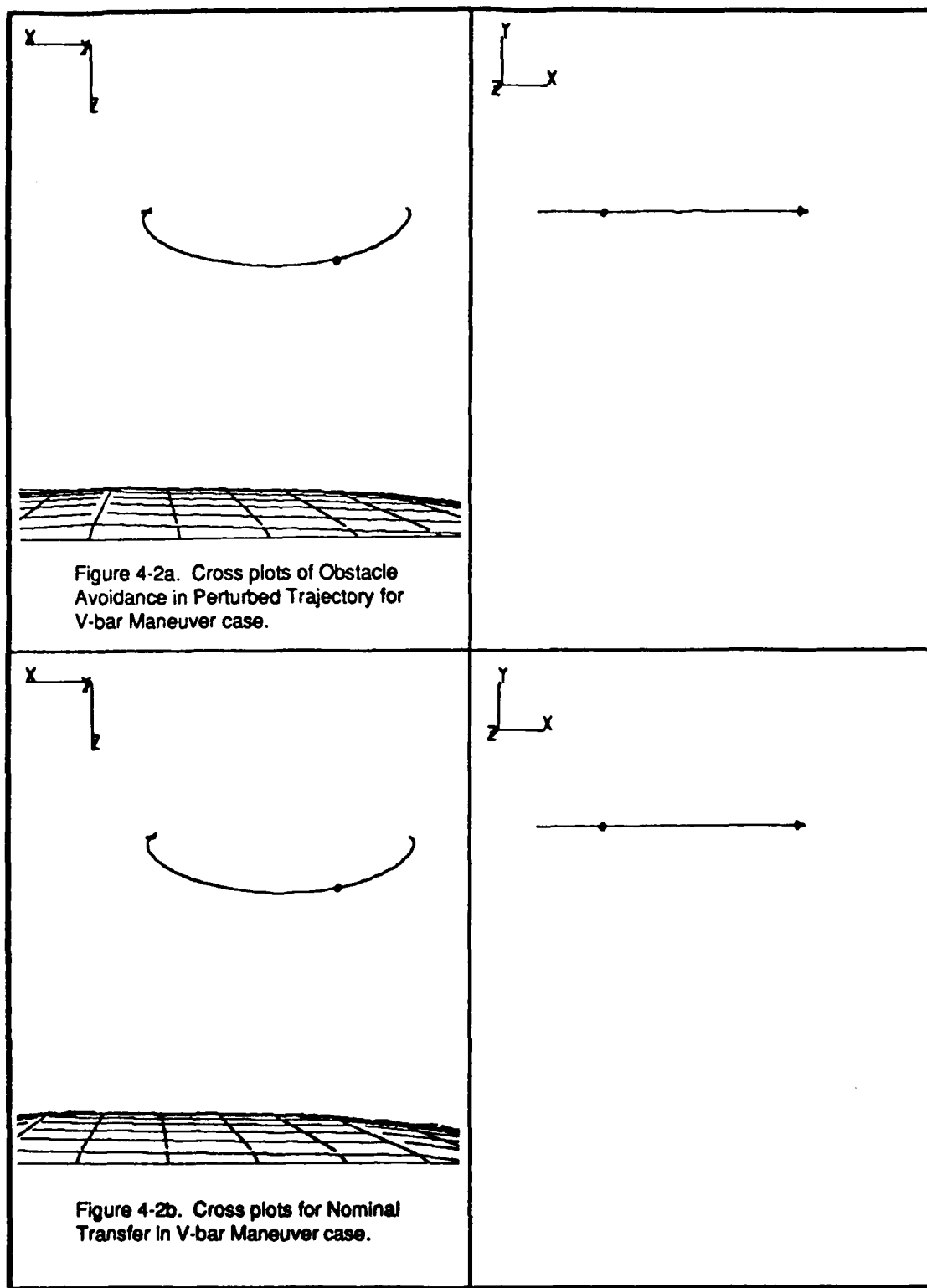
from the obstacle. The increase in fuel use due to obstacle accommodation is less than 5 percent of the nominal execution plan. (In all four test cases the obstacle was moving, but in the trace plots the obstacle has been artificially located at the point of collision between the intercepting vehicle and the chase vehicle on its nominal trajectory. In the Appendix and in the later section describing a multiple obstacle case, successive pictures taken of trajectory execution on the Space Systems Simulator show the actual separations between the obstacles and the chase vehicle.)

Case2: V-bar Maneuver--(-3000,0,0) to the origin. The chase vehicle is 3000 feet behind the target in this case. The V-bar approach is a well-known and preferred docking maneuver as it represents a very stable approach; if two vehicles in the same orbital plane have the same radius and same velocity, they will stationkeep without any jet firings. The nominal transfer time given to A\* was 5250 seconds.

Case	Initial Position* [ft]	Initial Velocity [ft/sec]	Nominal Time [sec]
2	-3000	-1.219 E -4	5250
	0	0	
	0	0	
	Fuel Use [lbs.] (No Obstacle)	Fuel Use [lbs.] (Obstacle)	Execution Time [sec] No Obstacle/Obstacle
	108.4	130.0	5250/6300

Table 4-2. Summary of Case 2

In the nominal case (no obstacle), the optimum transfer is expected to be performed over one orbit, corresponding to approximately 5400 seconds. To accomplish a pure phasing adjustment on a circular orbit, the chase spacecraft will put itself on a slightly eccentric orbit which will appropriately increase or decrease the orbital period. By doing so, the spacecraft will achieve the desired phasing adjustment while returning to the same altitude as the vehicle moves through a full orbit. As previously noted, a transfer problem



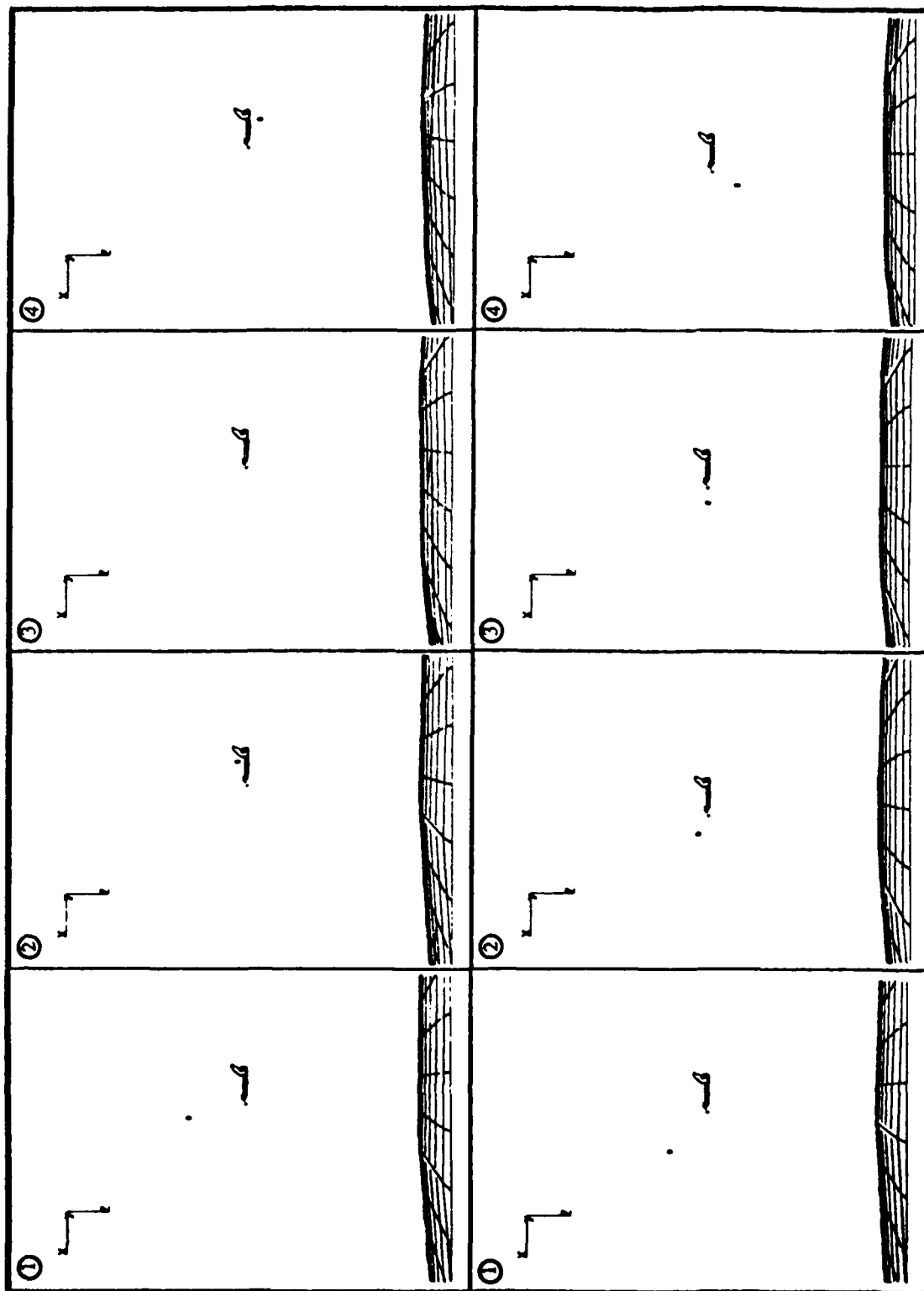


Figure 4-3. Comparison of Nominal (top) and Perturbed (bottom) Trajectories, V-bar Maneuver

## *Chapter 4. Testing*

which is unconstrained in time and seeks to minimize fuel will yield a solution with an infinite terminal time. In the four test cases presented, a time constraint of 7500 seconds was set, corresponding to just over one and a half orbits to preclude multiple orbit transfers. The nominal transfer time given to the search in this case is 5250 seconds and, as expected, the planner puts the chase vehicle on the slightly eccentric orbit to adjust for the phasing difference between the two vehicles.

In the obstructed case, the obstacle is targeted to intercept the chase vehicle with an out-of-plane trajectory. To avoid the obstacle, the planner puts the chase vehicle on a trajectory which delays the intercept point crossing. In Figure 4-2, there doesn't seem to be much difference between the nominal and perturbed trajectories; the chase vehicle's track seems to intersect the obstacle in both cases. When run in simulation with the moving obstacle (see Figure 4-3), it is clear that the chase vehicle avoids the collision by delaying the time at which it crosses through the intercept point. This illustrates a flexibility the planner has in obstacle avoidance. It has the capability to choose between or combine two approaches to perturbing the trajectory--physically moving the waypoints away from the location of the collision or moving the time of arrival at that point. In avoiding the obstacle the planner chooses to increase execution time by one time interval to 6300 seconds and accepts an increase in fuel expenditure of 20 percent.

It is important to note that a large portion of the increase in fuel expenditure is simply due to the increase in mission execution time. The majority of fuel use in each case is directly attributable to error regulation which is an integral part of the Digital Autopilot. As noted earlier, in closed loop execution, the DAP implements small perturbation jet firings along with the firings requested by the waypoint file to maintain the vehicle state vector to within a 3 foot sphere of the planned state. When this waypoint file was run open loop, suspending the DAP's error regulation, the resulting fuel use was 8.45 lbs as

## Chapter 4. Testing

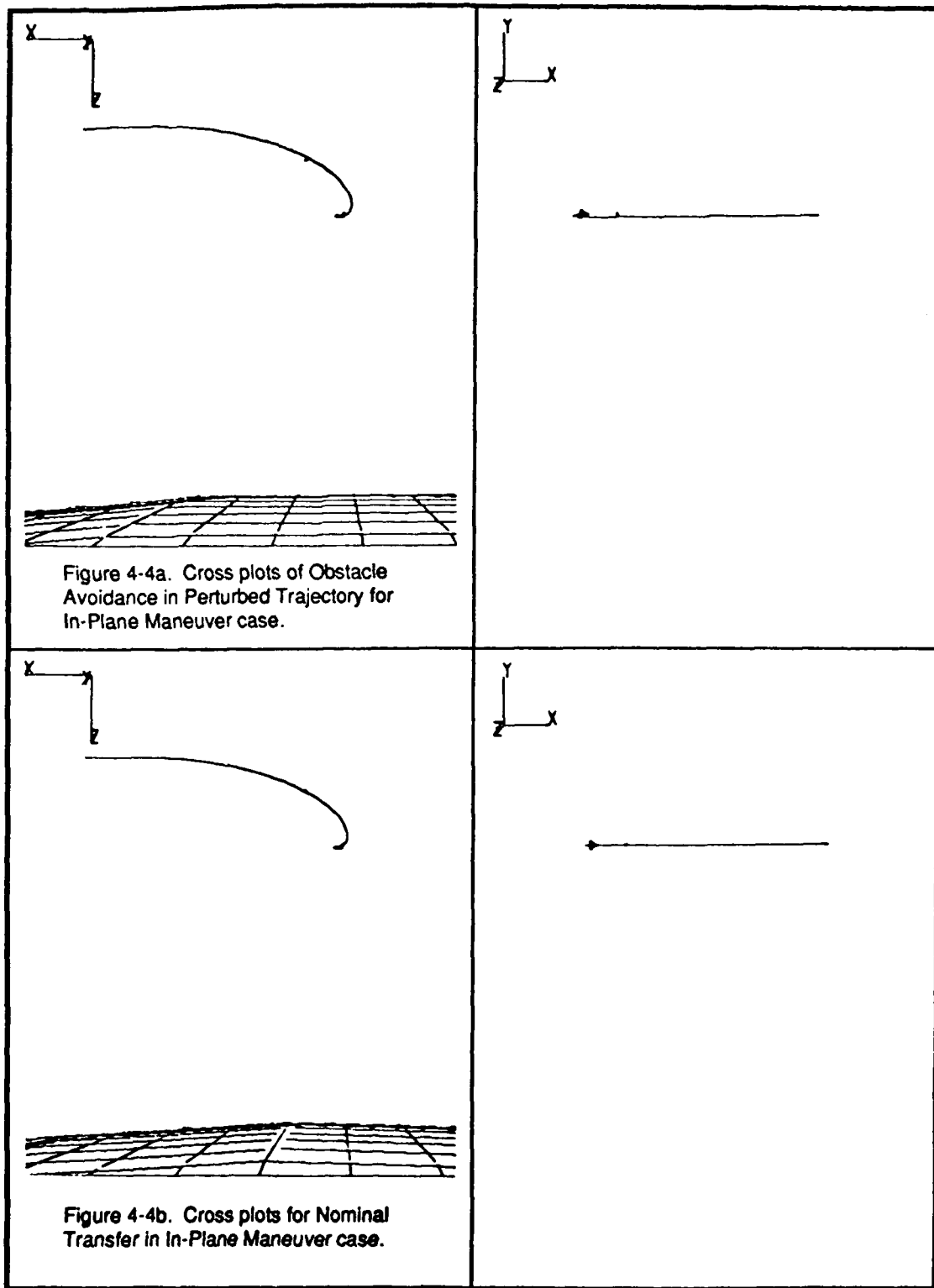
compared to 108.4 lbs for the closed loop execution. By increasing the length of this mission segment by 1050 seconds, the cost increases accordingly. The cost of error regulation is not taken into account in this implementation of the A\* algorithm and the planner is therefore blind to this cost. In future implementations, a penalty on mission execution time could be included in the cost function to represent this fuel expenditure inherent in the DAP.

**Case 3:** In-plane Maneuver--(3000,0,-1000) to the origin. In this case, the chase vehicle starts 3000 ft ahead and 1000 ft above the target vehicle. The initial displacements are still within the orbit plane, but while the nominal trajectory will remain in-plane, jet coupling makes out-of-plane maneuvers viable alternatives for obstacle avoidance.

Case	Initial Position* [ft]	Initial Velocity [ft/sec]	Nominal Time [sec]
3	3000	-1.74157	3000
	0	0	
	-1000	0	
3	Fuel Use [lbs.] (No Obstacle)	Fuel Use [lbs.] (Obstacle)	Execution Time [sec] No Obst/Obst
	69.6	72.3	3000/3000

Table 4-3. Summary of Case 3

In the nominal transfer, the planner accomplishes simple phasing and altitude adjustments. As such, the nominal plan reflects a balance between the two previous cases. Again by repeated execution with varying target times, a nominal time of 3000 seconds, which is slightly over half an orbit, is chosen. To accomplish the transfer, the chase vehicle decreases its tangential velocity, causing it to fall behind the target. As it falls behind the target, the chase vehicle also loses altitude and starts to regain tangential



## *Chapter 4. Testing*

velocity. The balance between altitude and tangential velocity bring the chase vehicle to the rendezvous point in just over half an orbit.

In the obstacle case, the intercepting vehicle approaches from within the orbit plane. (See Figure 4-4, Trace Plots for the In-Plane Maneuver.) Starting out behind and at the same altitude as the chase vehicle, the obstacle causes the chase vehicle to delay its descent and to make a small deviation out of plane. Because the intercepting obstacle remains in the original orbit plane and because jet coupling makes out-of-plane maneuvers cost effective, limited out-of-plane motion is introduced in the perturbed trajectory.

Had a variable time step been employed, the increase in cost could have been even further optimized. Instead, the planner must choose between arriving at the same time or entering a deceleration loop to account for an additional time step. The deceleration loop puts the vehicle on an arc which loops around the target location, tending to decrease the terminal velocity by using orbital mechanics. While such maneuvers decrease the terminal costs of the required maneuver, inefficiencies associated with quantized jet firings and discretized time intervals often make deceleration loops more costly. In this case, the planner selects the cheaper alternative and chooses to retain the same terminal time as the nominal case. To arrive at the same location in the same time, but from further away, the planner requests a higher closing velocity requiring a larger terminal impulse at the end of the transfer. To put the vehicle at the higher orbit and to null the higher closing velocity at the terminus, the vehicle expends approximately three additional pounds of fuel, about a 4% increase in cost.

Case 4: Out-of-plane Maneuver--(5000,1000,0) to the origin. (Note: The Appendix contains the steps in planning the nominal and obstacle trajectories for this case.) The chase vehicle, in this problem, starts out 5000 feet ahead and 1000 feet to the right of

## Chapter 4. Testing

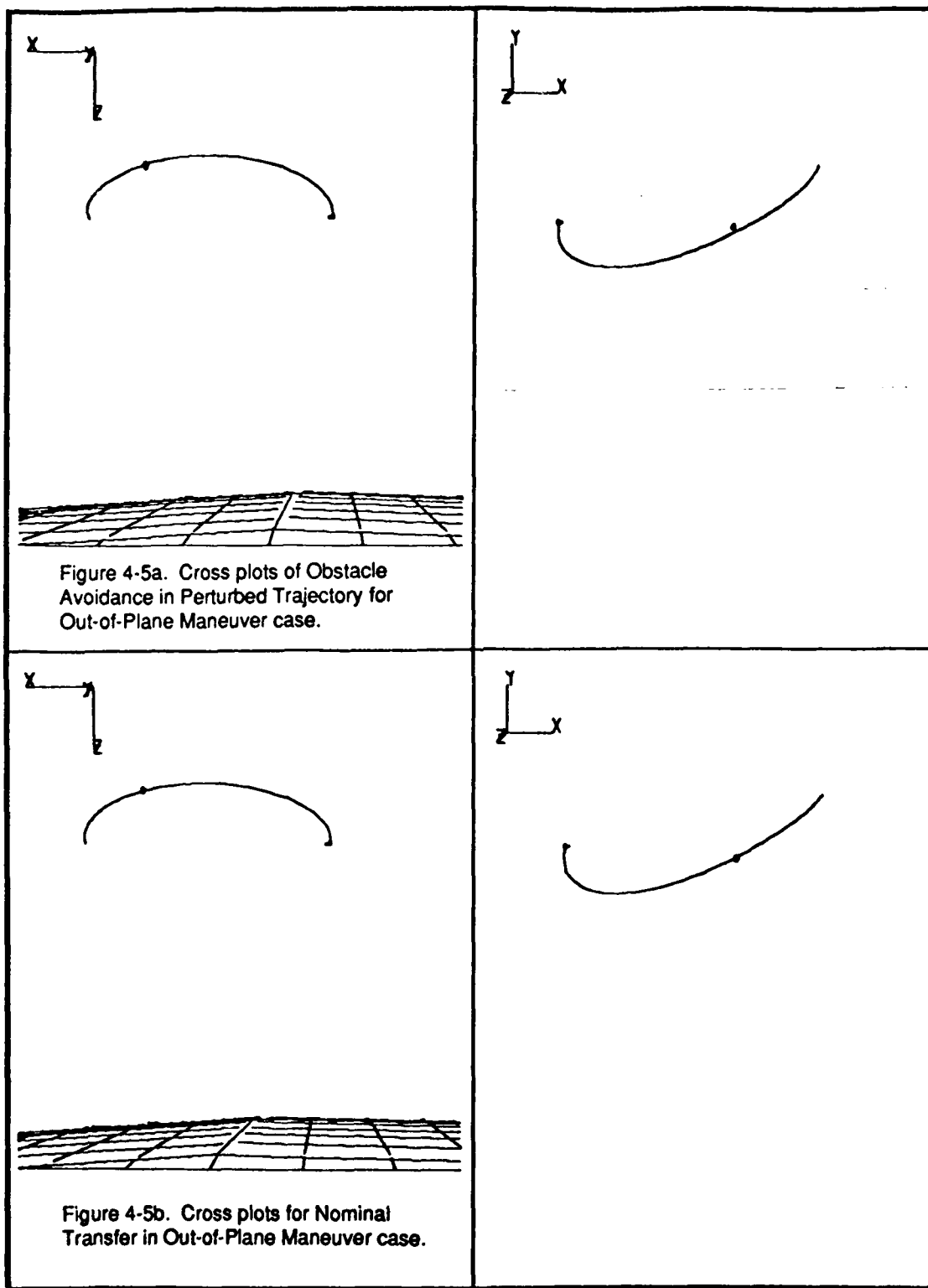
the target (looking along the velocity vector). The chase vehicle is located about the same distance from the center of attraction as the target, but is on a different orbit plane. Because the nature of the solution necessarily includes a plane change, it requires a significant fuel expenditure, but this also provides many options for avoiding obstacles and for optimizing the path through multiple jet firings. The nominal time for execution is 4250 seconds, which is just over three-quarters of an orbit.

Case	Initial Position* [ft]	Initial Velocity [ft/sec]	Nominal Time [sec]
4	5000	-3.426 E -4	4250
	1000	-1.5933 E -3	
	0	0	
	Fuel Use [lbs.] (No Obstacle)	Fuel Use [lbs.] (Obstacle)	Execution Time [sec] No Obst/Obst
	125.1	131.0	4250/4250

Table 4-4. Summary of Case 4

The nominal case is again a balance of two effects. The phasing change with no altitude change is best accomplished at a multiple of  $2\pi$  radians, while the out of plane adjustment is best completed at odd multiples of  $\pi/2$  radians. The nominal time picked for this case is slightly over  $3\pi/2$  radians, a choice which allows for most of an orbit to accomplish the phasing adjustment while retaining most of the out of plane motion. A transfer corresponding to  $5\pi/2$  radians is also a good candidate given the problem statement in which mission execution time is constrained to be less than 7500 seconds or approximately  $3\pi$  radians. Nominal time selection, however, was limited to 6000 seconds, to bias solutions toward trajectories that would execute within one orbit. It is important to note that the planner still could have produced solutions corresponding to the  $5\pi/2$  transfer, but A\* viewed them as higher cost alternatives to the  $3\pi/2$  trajectory which more closely matches the nominal time. In certain cases, the tendency of this planner to prefer solutions near to the nominal time can be deleterious to the optimizing efforts of the trajectory planner. In future implementations, this can be avoided by altering the node





## **Chapter 4. Testing**

generation algorithm early on in the search. (For further discussion on alternate node generation schemes, see the Conclusions section.)

In the nominal trajectory (see Figure 4-5), the planner generates three impulses early in the transfer to shape the motion of the vehicle in the  $y$  (out-of-plane) direction. The chase vehicle follows motion that is only slightly different than the sinusoidal motion (relative to the orbit plane) expected in unforced Keplerian motion, while successfully managing orbital energy to null the initial in-plane displacements as well.

In the obstacle case (see Figure 4-5), the chase vehicle's track in the  $y$ -direction is very similar to the nominal trajectory. Instead of introducing large out-of-plane perturbations, the planner avoids the obstacle by making adjustments in the  $x$  ( $\bar{v}$ ) and  $z$  ( $\bar{r}$ ) directions. The manner of accommodating the obstacle is similar to each of the other cases. The planner chooses a trajectory which delays motion toward the target until the obstacle moves past the interception point and then completes the approach to the goal state. Because it is again forced to choose between arriving at the nominal time or extending for an entire time interval ( $\Delta T$ ), the planner, in choosing the lower cost alternative, produces a trajectory with a higher closing velocity to complete the transfer at the nominal time. The consequential increase in fuel use is still, however, under 5% of the nominal transfer cost.

### **MULTIPLE OBSTACLES**

To demonstrate the flexibility afforded by the trajectory planner, a case with three moving obstacles was posed, using the problem statement of Case 2. The successive frames in Figure 4-6, Obstacle Interception of Nominal Trajectory, show the direct

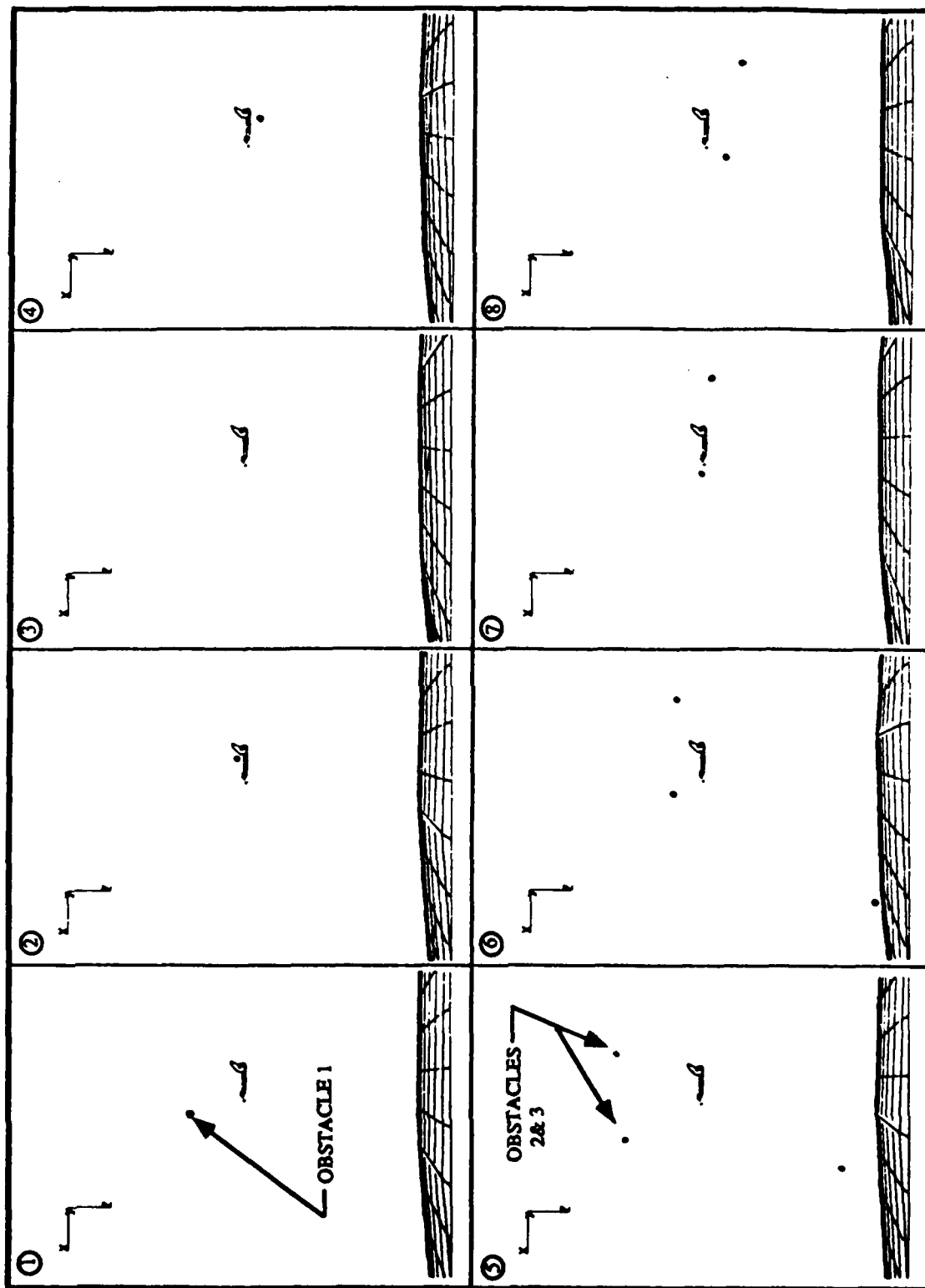


Figure 4-6. Obstacle Interception of Nominal Trajectory

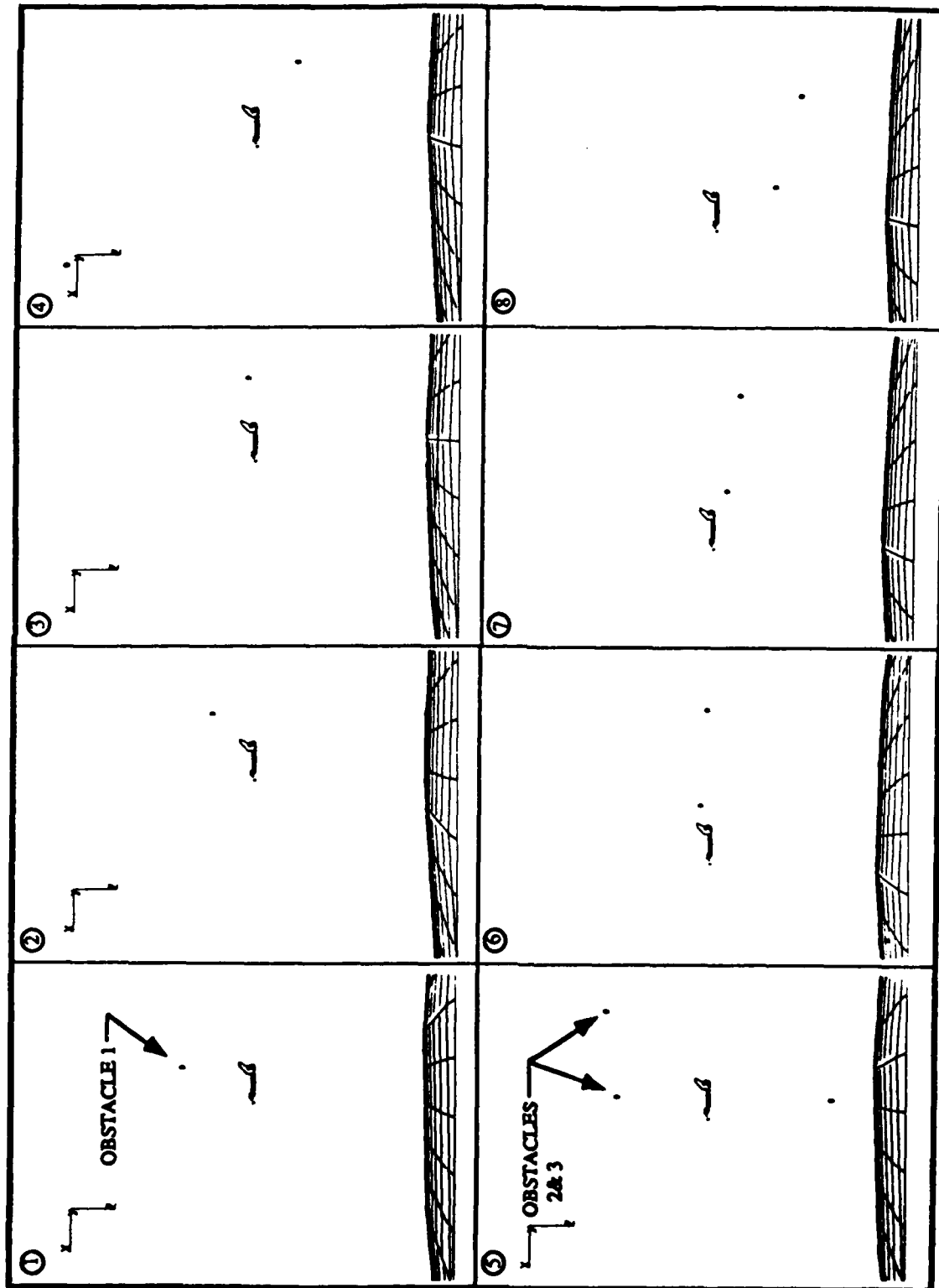


Figure 4-7. Obstacle Avoidance in Perturbed Trajectory

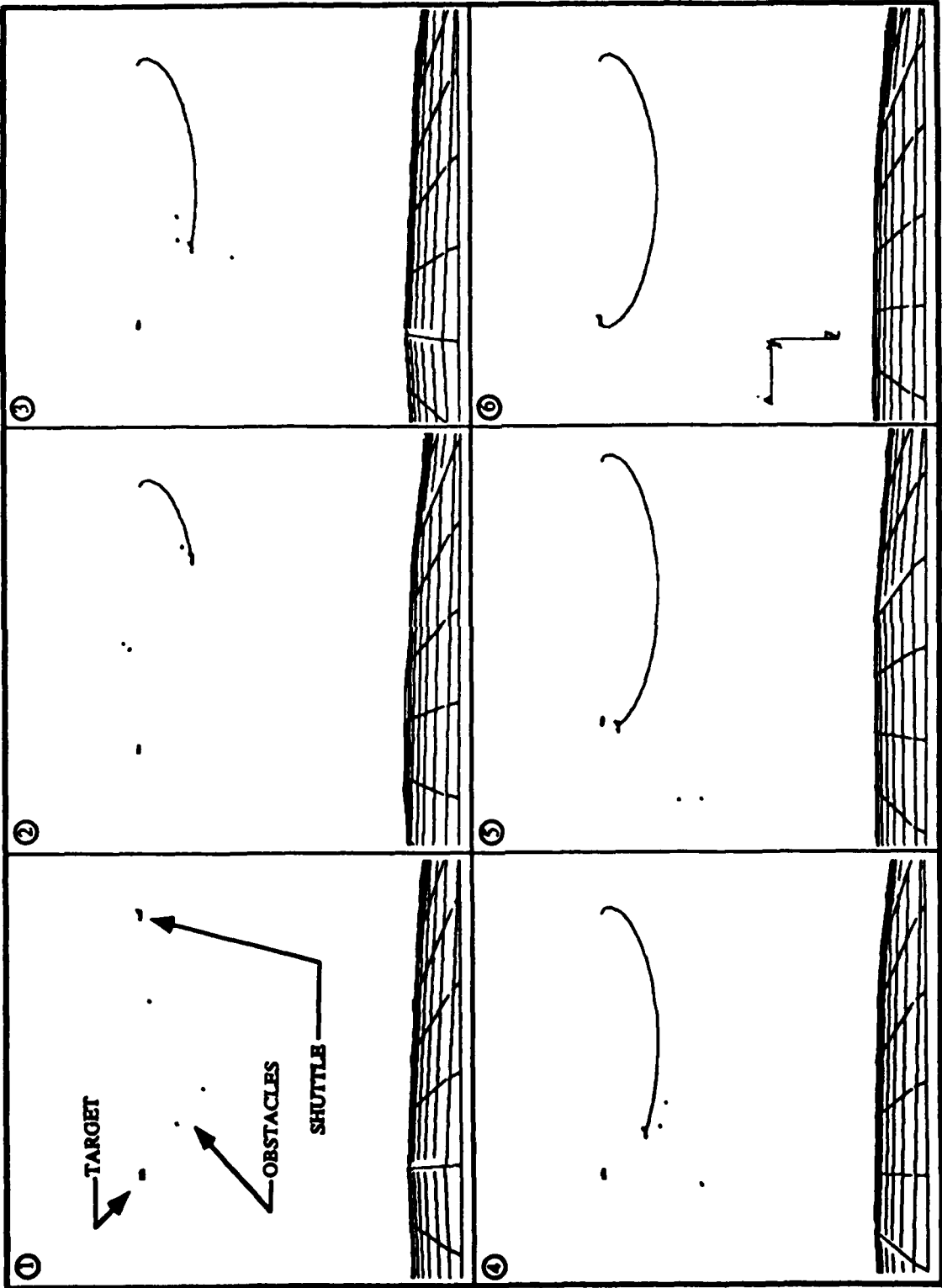


Figure 4-8. Trajectory Summary for Multiple Obstacle Case

## ***Chapter 4. Testing***

collision of the chase vehicle with Obstacle 1 (see frame 3, Figure 4-6) in the nominal trajectory. Later in the nominal trajectory, Obstacles 2 and 3 cross forward and aft of the chase vehicle simultaneously. Small alterations in flight path to avoid Obstacle 1, made without considering the other obstacles, will cause collisions with Obstacles 2 and 3. The A\* planner accounts for all three obstacles and generates a trajectory which takes the chase vehicle through the interception point in advance of the obstacles.

In generating the interception points, Obstacle 1 was given initial conditions which would place it in the chase vehicle's nominal trajectory. With the information on the first obstacle, the A\* algorithm yielded a corrected trajectory which altered the path enough to avoid the obstacle. Obstacle 2 was then assigned initial conditions which would cause it to collide with the chase vehicle in its perturbed path. Given the additional information on the second obstacle, A\* planned a third trajectory, optimizing its solution and avoiding both obstacles. This process was repeated for a third obstacle to stress the algorithm's capability

Figure 4-7 shows successive frames illustrating the chase vehicle's final perturbed trajectory as it avoids the three obstacles. (The overall transfer is shown in Figure 4-8.) Interestingly, the increase in fuel was only half of that required by the single obstacle case discussed above (fuel consumption for three obstacle case, 116.1 lbs). This result is attributed to the inadmissibility of the heuristic estimate which can cause an A\* search algorithm to produce a non-optimal solution. This effect will be discussed in greater detail in the Conclusions.

### **COMPARISON TO A TWO-IMPULSE TRAJECTORY SOLVER**

To create a basis for comparison, two of the test cases were run on a program which solved the Clohessy-Wiltshire Equations for the two-impulse solution which would

## Chapter 4. Testing

correspond to the same transfer time as the A\* solution. While we make no claim as to the global optimality of the solutions from this particular trajectory planner, we observe a favorable comparison between the CW (Clohessy-Wiltshire) Solver and the A\* Trajectory Planner performance.

Case	Fuel Use [lbs.]	Execution Time [sec]
1	163.4	3750
4	133.7	4250

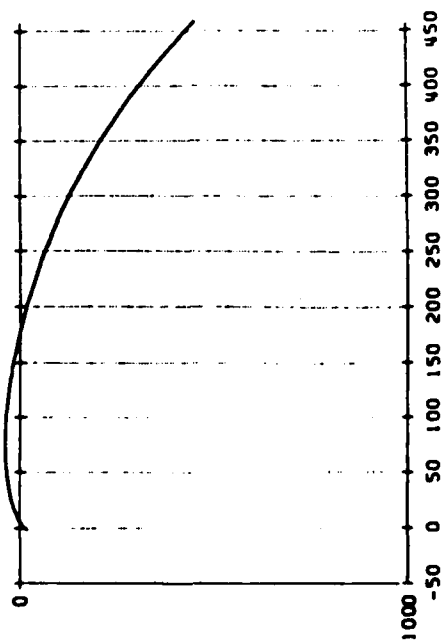
Table 4-5. Summary of CW Solver Performance.

In Case 1 (the altitude change maneuver), the two-impulse trajectory specified by the CW Solver requires about the same fuel use as (approximately 1% less) the multiple segment trajectory planned by A\*. However, in the out of plane transfer defined by Case 4, the A\* trajectory is approximately 6% better than the CW trajectory. While it is possible to find more efficient two-burn trajectories by accomplishing a simple gradient search on transfer times, it is evident that this Trajectory Planner provides reasonable trajectories while affording the flexibility to accommodate significant, unplanned obstacles which may be encountered during execution of a nominal trajectory.

## UNPLANNED DISTURBANCE

One of the disturbances which a more realistic simulation includes is the effect of the J2 term in the Earth's gravitational potential. This term corresponds to the earth's oblateness. In orbits which are not in the equatorial plane, this causes precessing or recessing of the line of nodes depending on the inclination of the orbit [8]. The Trajectory Planner is unaware of this disturbance and therefore cannot account for it, hence the actual trajectory will tend to deviate from the planned trajectory. The Execution Manager, through

4-9a.



4-9b.

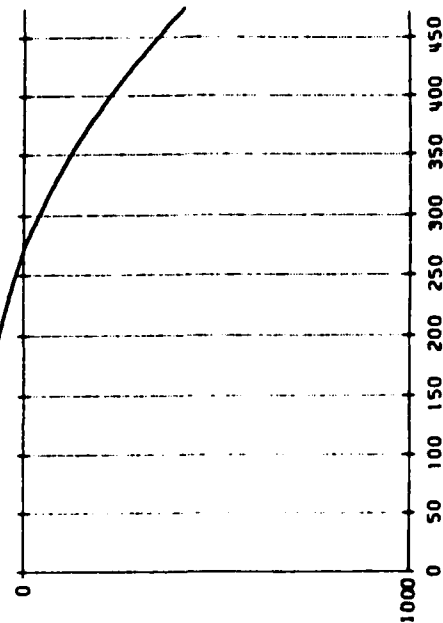
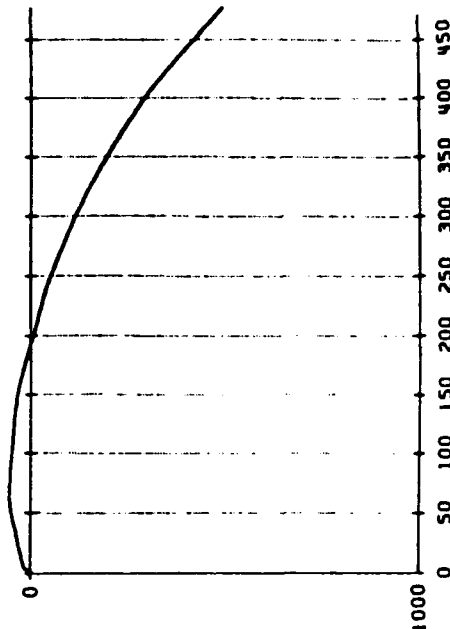


Figure 4-9. Termination of V-bar Transfer

- a. Open Loop execution of Waypoint File
- b. Open Loop execution with J2 effects
- c. Closed Loop execution with J2 effects

4-9c.





## *Chapter 4. Testing*

the DAP, is able to regulate the error to keep the state vector near to the planned trajectory and arrive at the specified goal state.

Figure 4-9a shows the Open Loop nominal execution which gets the chase vehicle close to the goal. Two numerical inaccuracies decrease the positional accuracy of the transfer. The more obvious, but less significant source, is computational inaccuracy due to round off and linearization in the algorithm. The other source of terminal error is the implementation of the jet firings. In planning the chase vehicle's trajectory, impulsive velocity changes are assumed. In simulation, the jet firings are not impulsive and actually occur over finite intervals.

Figure 4-9b shows the Open Loop execution with the J2 effect. There is a notable error at the termination of the transfer (approximately 125 feet). Figure 4-9c shows the corrections which the DAP is able to accomplish, bringing the chase vehicle to exactly match the specified goal conditions (zero displacement and zero velocity).

In practice, all trajectories would be implemented by an error-regulated closed loop controller, but this exercise serves to demonstrate the effectiveness of the DAP in error removal due to disturbances which may cause deviations from a planned trajectory.

## **5 CONCLUSIONS**

### **SUMMARY**

This implementation of the A\* node search technique illustrates the concept of an optimal search algorithm (OSA) within the framework of a Trajectory Planner. By generating an algorithm which is computationally efficient, the idea of a real-time planning system that can replan in the face of evolving environmental constraints is validated.

The OSA concept that was developed in this effort is built on the concept of a neighboring optimal controller which starts with a nominally optimal control policy that evolves with the introduction of constraints. This planner requires as an input, the nominal time, representing the nominally optimal trajectory between the initial and terminal states and views intercepting obstacles as physical constraints on the state vector. The other inputs to the search affect the performance of the OSA; these include the weight on the variation of the quanta vector and the number of time intervals in the nominal transfer, which sets the length of a time step.

The capability of the search to avoid obstacles by either changing the time of passage through a particular location or by changing the path traced out by the trajectory has been demonstrated. Although moving obstacles were used in these cases, it is clear that obstacles which are stationary relative to the target, such as solar arrays or other structural appendage, can also be avoided by altering just the physical trajectory.

Favorable comparisons were also made to trajectories produced by a CW Solver which planned two-impulse transfers. This demonstrates that the solutions provided by

## *Chapter 5. Conclusions*

this multiple segment trajectory planner are viable, and can be more efficient than solutions provided by a planner which was based on two-impulse trajectories. The combined advantages of obstacle avoidance and optimization are significant developments for a proximity operation controller.

Some difficulties and limitations of this approach became apparent in the development of this algorithm. The primary hindrance to implementation of an optimal search algorithm is likely to be computer memory requirements. The OPEN and CLOSED lists can become extremely lengthy in cases where the cost gradient is very small. Pruning methodologies are available which could ease memory requirements for future implementations of the A\* search [11].

The A\* algorithm used penalties on the nodes associated with collision states to avoid contact between the chase vehicle and intercepting obstacles. As discussed in Chapter 4, the nodes which represent collisions with obstacles could have been pruned and removed from consideration. This would remove candidate nodes from the theoretical search space, apparently sacrificing a degree of fuel optimality. However, since these nodes represent collisions and thus are unacceptable to a viable solution string, there is no need to maintain these as candidates, no matter how fuel optimal they may appear to the search algorithm. Further, retaining these nodes has the deleterious effect of lengthening the OPEN list; this, then, is an area where a minor modification could increase search speed and decrease memory requirements.

Significant limitations on trajectory optimization were realized because of the grid assignment concept employed. By using Dynamic Grid Assignment, the search was confined to a considerably reduced subset of the trajectories which reflect the full range of possible effector combinations. While the A\* algorithm may have picked the best solution

## *Chapter 5. Conclusions*

from that subset there was no guarantee that it was the optimal solution overall. On the other hand, by virtue of the way that the grid was generated, all the nodes are guaranteed to be physically realizable. This concept therefore constitutes an implicit pruning mechanism of sorts. Employment of concepts contained in an alternative approach to jet selection [14] could lead to useful developments in expanding variations on the quanta vector. (This will be discussed in the next section on topics for further research.) While a degree of optimality was sacrificed in employing the Dynamic Grid Assignment concept, the goal of creating a workable optimal search algorithm for the Trajectory Planner was attained.

The most significant degradation to optimality was caused by inadmissible heuristics. In the results section, the effects of the inadmissible heuristic were noted. Throughout the search for a solution string, the total cost (FCOST) should monotonically increase. Instead, FCOST sometimes oscillated during the search. This happened when the heuristic estimate (HCOST), the cost of the two-impulse trajectory from an intermediate node to the goal, was not necessarily smaller than the cost of all other trajectories. If a more efficient multiple-impulse trajectory was expanded, it decreased FCOST in successive generations. This ultimately affected optimality by causing the search to overlook optimal nodes in favor of other nodes which were not as promising, because the estimate in the optimal direction was too high. While the heuristic estimate could simply have been scaled to meet the admissibility requirement this would have decreased the aggressiveness of the heuristic, and would have decreased the cost gradient. As noted above, this would have consequently diminished search speed.

Nevertheless, an algorithm which successfully avoids obstacles and provides a capability of accounting for evolving constraints was created. Many avenues for further research exist which can lead to more effective expansions of successor nodes, thereby

increasing optimality, as well as increasing search efficiency, consequently decreasing the time required to replan.

## RECOMMENDATIONS FOR FUTURE WORK

Variation on Quanta. The choice between providing a wide enough range of variation and creating too many nodes per generation will always be a design dilemma. Too small a range of variation creates a problem for the search, because all of the variations are essentially the same node. In this implementation, this can occur when the velocity change requested is very small, which often happens at either transfer terminus. The variation on the quanta vector is a percentage of the particular component of the quanta vector. For example if the quanta vector was (100, 70, 80) a 20 % variation might be (80, 56, 64) which is significantly different. If, however, the quanta vector were (5, 1, 1), the variations are extremely limited; each component of the quanta vector in a 20% variation could at most be varied by 1 quanta, corresponding to a 80 millisecond jet firing, which results in a velocity change of 0.0105 ft/sec for the the typical Space Shuttle effector. The variations on the latter quanta vector would clearly not cover as wide a range as variations on the former.

While increasing the weighting percentage of the variation may seem to be a quick answer, this policy would only be marginally effective. If, for example, the weight was increased to 50%, the (5, 1, 1) vector would still be varied at most by 2 or 3 quanta. Further, there are additional drawbacks to this quick answer. As the percentage variation increases, so does the granularity of the control effort, causing two potential difficulties. If there happens to be a local minimum in the cost function which resides between the coarse variations created by a large weighting percentage, this local minimum will never be seen and a degree of optimality might be sacrificed.

## *Chapter 5. Conclusions*

The second potential difficulty arises from the way that the variations generate the successor nodes. After creating the variations on the quanta vector, these on-times are applied to the corresponding jets. This produces variations in the vehicle's velocity space. It follows that large variations in the quanta vector, which directly correspond to jet-on times, will produce large variations in velocity change requests. The difficulty arises when the varied velocity change request is not best implemented by the combination of jets selected for the nominal velocity change. If a different set of effectors leads to more efficient fuel use, the DAP will choose that combination of thrusters. In that case, the quanta vector which was used to calculate actual cost will no longer be an accurate cost measure, and again optimality claims are sacrificed.

A different approach to developing the quanta vector which relies on an alternative to the Linear Programming Jet Select algorithm was briefly investigated. Glandorf [14] states that in the  $N$  degree of freedom effector selection problem can be viewed as picking from adjacent  $N+1$  sided polyhedra. The volume contained by each polyhedron represents a velocity change in  $N$ -space and each leg represents an activity vector. A given velocity change request will lie within the volume of one pyramid. The legs of the pyramid correspond to a particular activity vector or jet cluster and the linear combination of the vectors which these legs represent is the quanta vector which is required for that nominal velocity change request. As before, the objective is to create useful variations on the nominal velocity change vector. This is accomplished by picking other vectors which reside in the subspace represented by the polyhedron. In addition, Glandorf's algorithm identifies adjacent polyhedra corresponding to different sets of effectors which can be used to generate additional variations which are significantly different but guaranteed to be optimal under the current environmental conditions and vehicle status. The design dilemma still remains, however--the designer must decide how many adjacent polyhedra to use, and

## *Chapter 5. Conclusions*

how to discretize the space within each polyhedron. Too many nodes will unnecessarily burden the search algorithm, but too few will, at best, lead to a non-optimal solution, and, at worst, yield no solution.

Variation on Time Step Size. A reduction in the magnitude of the velocity change required to complete a given mission can be realized by allowing the terminal time to vary [8]. In the results section, Case 3 noted that the planner was forced to choose between arriving at the same terminal time as the nominal trajectory or extending by one time interval. While the optimal terminal time was probably somewhere in between, the planner was not offered that choice.

By creating a simple gradient algorithm, the terminal time could be varied to minimize the cost to complete the mission. Indeed, this type of optimization could be beneficial at all time steps in the transfer. Consider, for example, the no-firing interval that occurs midway through the trajectory which is the best option if it lasts one and a half time steps but is not optimal if it lasts one or two time steps. By allowing a variable time step, this local cost minimum could be identified and exploited without increasing the number of nodes generated to produce a solution trajectory.

Extension of Dynamics Model. A logical extension of the preliminary developments in this effort (which only account for the three translational degrees of freedom) is the description of the attitude dynamics of the various vehicles used by the planner. Each vehicle--the chase, the target, and the obstacles--has six degrees of freedom and the interaction of vehicle attitudes, if accounted for could be exploited to achieve greater fuel economy. This effort has addressed an optimized trajectory planning scheme in only the translational components, treating each spacecraft as a uniform sphere generated around

## *Chapter 5. Conclusions*

a center of mass. Great advantages in fuel can be imagined when considering the close-in maneuvering necessary for autonomous docking in servicing or retrieval missions.

Other improvements to the dynamics model used by the planner might include modelling gravity gradient torque, J2 effects, and atmospheric drag. Unmodelled, each of these effects would be treated as an unplanned disturbance and nulled by the closed loop regulation of the DAP. Instead, if these environmental effects were reflected in the orbital dynamics, the planner might be able to use these effects to make certain solution trajectories feasible. For example, the planner may be able to use atmospheric drag to null terminal velocity in a docking maneuver which would have previously been impossible without plume impingement of the target [15]. J2 effects may aid in making phasing corrections while gravity gradient torque may help with attitude adjustments.

Cooperative Control. The Cooperative Control concept [16] accomplishes proximity operations using two active vehicles. In the developments described here, the chase vehicle is active while the target vehicle is passive. The nodes generated by the planner are target relative positions which result from chase vehicle control combinations. If, instead, both spacecraft were active and able to coordinate closing motion, the effector sets on both vehicles could be accessed. It follows that the most effective thrusters would be used to accomplish the required maneuvers. The challenge would again be a familiar one--how to limit the control combinations used in expanding successive generations to a manageable node space.



## **APPENDIX A**

### **Development of Solutions for the Out-of-Plane Maneuver**

The initial conditions of the chase vehicle, the target state, the nominal time for the transfer, stepsize, variation weighting factor, used to construct the nominal path for the specified transfer are as follows:

Initial conditions:

Displacement: (5000, 1000, 0) feet, corresponding to 5000 feet ahead and 1000 feet to the right of the target (looking along the velocity vector).

Velocity: (-3.426E-4, -1.5933E-3, 0) ft/sec (relative to the target).

Target state conditions:

Displacement: (0,0,0) feet

Velocity: (0,0,0) ft/sec

Nominal time:

4250 seconds

Stepsize:

5 equal time increments in the transfer

Variation Weighting Factor:

20% variations on nominal

The output of the planner, given this information, is a waypoint file shown in Figure A-1. Three cross plots are included which graphically display this data (see Figure

A-2). In creating the obstacle case, the data in the waypoint file is used to pick a time when an intercepting obstacle will "target" the chase vehicle in its nominal trajectory. In this case, 1700 seconds is chosen. The nominal trajectory puts the chase vehicle at (1142.10, -166.114, -971.422) feet from the target. Using the CW Solver (which simply solves the Clohessy-Wiltshire equations for the velocity required to get from one point to another in a specified time), we arrive at the initial conditions for the intercepting obstacle:  
Initial Displacement: (4000, -500, 0), Initial Velocity: (0.475312, 0.461407, 0.214931) ft/sec.

Given this information on the obstacle, the A\* planner calculates a new trajectory which will avoid collision with the obstacle at 1700 seconds, and indeed, at any time in the transfer. In the planning process, collision with an obstacle incurs a large penalty, causing the search algorithm to stay away from expanding nodes which would allow intersecting paths. The new perturbed trajectory is given in Figure A-3, and graphically displayed through cross plots in Figure A-4.

The remaining step is the simulation of this result. Transferring the waypoint file to the Space Systems Simulator, the DAP implements the trajectory. A sequence of slides is presented in Figure A-5 showing the chase vehicle clearly avoiding the moving obstacle and successfully terminating the mission by docking with the target. (The MET display shows Mission Elapsed Time. This clock is zeroed at the beginning of the transfer, and is thereby useful as a relative time index in interpreting vehicular activity.)

Figure A-1. WAYPOINT FILE FOR NOMINAL OUT OF PLANE MANEUVER

X	Y	Z	VX	VY	VZ	TIME	FLAG
5000.00	1000.00	.000000	.266660	.209281	.344019	.000000	1
5014.51	1010.12	-21.7385	.216186	.127973	.380303	.60.0000	0
5025.84	1015.34	-45.5995	.160784	4.604432E-02	.414744	120.000 0	
5033.71	1015.64	-71.4674	.100722	-3.610758E-02	.447172	180.000 0	
5037.84	1011.01	-99.2168	3.629176E-02	.118084	.477432	240.000 0	
5037.99	1001.48	-128.713	-3.219435E-02	.199488	.505376	300.000 0	
5033.91	987.096	-159.813	.104404	.279925	.530869	360.000 0	
5025.39	967.920	-192.366	.179988	.359004	.553788	420.000 0	
5012.25	944.050	-226.214	.258578	.436342	.574021	480.000 0	
4994.31	915.601	-261.193	.339794	.511563	.591469	540.000 0	
4971.43	882.712	-297.133	.423242	.584304	.606049	600.000 0	
4943.48	845.542	-333.860	.508517	.654210	.617690	660.000 0	
4910.38	804.270	-371.196	.595206	.720544	.626335	720.000 0	
4872.04	759.098	-408.959	.682888	.784181	.631942	780.000 0	
4828.42	710.245	-446.967	.771138	.843615	.634484	840.000 0	
4820.63	701.761	-453.313	.789666	.769598	.721272	850.000 1	
4770.24	653.921	-496.553	.890063	.824434	.719485	910.000 0	
4713.84	602.909	-539.581	.989969	.875272	.714210	970.000 0	
4651.46	548.973	-582.189	1.08890	.921865	.705470	1030.00 0	
4583.20	492.3	-624.169	1.18637	.963986	.693309	1090.00 0	
4509.14	433.358	-665.319	1.28191	1.00143	.677785	1150.00 0	
4429.42	372.500	-705.438	1.37507	1.03402	.658974	1210.00 0	
4344.19	309.406	-744.332	1.46537	1.06160	.636967	1270.00 0	
4253.64	245.011	-781.812	1.55240	1.08402	.611871	1330.00 0	
4157.97	179.478	-817.697	1.63572	1.10119	.583807	1390.00 0	
4057.43	112.975	-851.812	1.71493	1.11302	.552911	1450.00 0	
3952.27	45.9719	-883.992	1.78965	1.11945	.519334	1510.00 0	
3842.77	-21.2701	-914.082	1.85951	1.12045	.483239	1570.00 0	
3729.23	-88.311	-941.934	1.92418	1.11601	.444799	1630.00 0	
3611.98	-155.064	-967.415	1.98334	1.10617	.404203	1690.00 0	
3592.10	-166.14	-971.422	1.99483	1.03994	.413587	1700.00 1	
3470.74	-228.58	-994.950	2.04946	1.02400	.370379	1760.00 0	
3346.29	-288.595	-1015.83	2.09794	1.00309	.325375	1820.00 0	
3219.12	-348.331	-1033.96	2.14004	.977312	.278793	1880.00 0	
3089.62	-406.777	-1049.26	2.17556	.946796	.230859	1940.00 0	
2958.19	-461.854	-1061.64	2.20431	.911689	.181805	2000.00 0	
2825.24	-515.592	-1071.06	2.22617	.872160	.131869	2060.00 0	
2691.19	-566.429	-1077.46	2.24103	.828401	.8.129379E-02	2120.00 0	
2556.45	-614.719	-1080.81	2.24880	.780625	-3.032443E-02	2180.00 0	
2421.47	-660.028	-1081.09	2.24947	.729062	2.079225E-02	2240.00 0	
2286.66	-702.136	-1078.31	2.24301	.673964	7.180792E-02	2300.00 0	
2152.45	-740.839	-1072.48	2.22948	.615596	.122475	2360.00 0	
2019.26	-775.948	-1063.63	2.20892	.554244	.172549	2420.00 0	
1887.52	-807.294	-1051.79	2.18144	.490203	.221785	2480.00 0	
1757.63	-834.725	-1037.03	2.14717	.423784	.269946	2540.00 0	
1736.19	-838.906	-1034.30	2.14081	.412506	.277852	2550.00 1	
1608.97	-861.602	-1016.22	2.09884	.343722	.324463	2610.00 0	
1484.45	-880.120	-995.391	2.05048	.273270	.369501	2670.00 0	
1363.03	-894.368	-971.915	1.99597	.201493	.412746	2730.00 0	
1245.06	-904.279	-945.902	1.93558	.128740	.453990	2790.00 0	
1130.87	-909.804	-917.480	1.86958	-5.536124E-02	.493032	2850.00 0	
1020.81	-910.917	-886.786	1.79832	1.828559E-02	.529682	2910.00 0	

Figure A-1. (Continued)

915.174 -907.612 -853.969 -1.72212 9 184375E-02 563764 2970.00 0  
 814.247 -899.905 -819.189 -1.64137 164956 595111 3030.00 0  
 718.293 -887.833 -782.614 -1.55644 237269 623573 3090.00 0  
 627.549 -871.456 -744.421 -1.46776 308431 649009 3150.00 0  
 542.228 -850.851 -704.796 -1.37576 378097 671299 3210.00 0  
 462.516 -826.121 -663.930 -1.28087 445929 690332 3270.00 0  
 388.572 -797.383 -622.023 -1.18357 511599 706017 3330.00 0  
 320.526 -764.778 -579.277 -1.08432 574787 718278 3390.00 0  
 309.767 -758.979 -572.085 -1.06762 585055 719985 3400.00 1  
 248.731 -722.064 -528.623 -966709 644963 728177 3460.00 0  
 193.779 -681.646 -484.775 -864900 701742 732837 3520.00 0  
 144.951 -637.923 -440.753 -762689 755117 733944 3580.00 0  
 102.255 -591.105 -396.772 -660571 804830 731490 3640.00 0  
 65.6708 -541.421 -353.045 -559043 850640 725489 3700.00 0  
 35.1482 -489.111 -309.784 -458596 892324 715970 3760.00 0  
 10.6079 -434.429 -267.199 -359718 929681 702978 3820.00 0  
 -8.05891 -377.639 -225.495 -262888 962528 686576 3880.00 0  
 -20.9891 -319.019 -184.876 -168576 990708 666845 3940.00 0  
 -28.3475 -258.851 -145.538 -7.72396E-02 1.01408 643879 4000.00 0  
 -30.3262 -197.427 -107.673 1.067880E-02 1.03254 617791 4060.00 0  
 -27.1430 -135.046 -71.4634 9.475233E-02 1.04599 588706 4120.00 0  
 -19.0410 -72.0103 -37.0852 174574 1.05436 556766 4180.00 0  
 -6.28697 -8.62513 -4.70544 249755 1.05762 522126 4240.00 0  
 .000000 .000000 .000000 .000000 .000000 .000000 4250.00 1

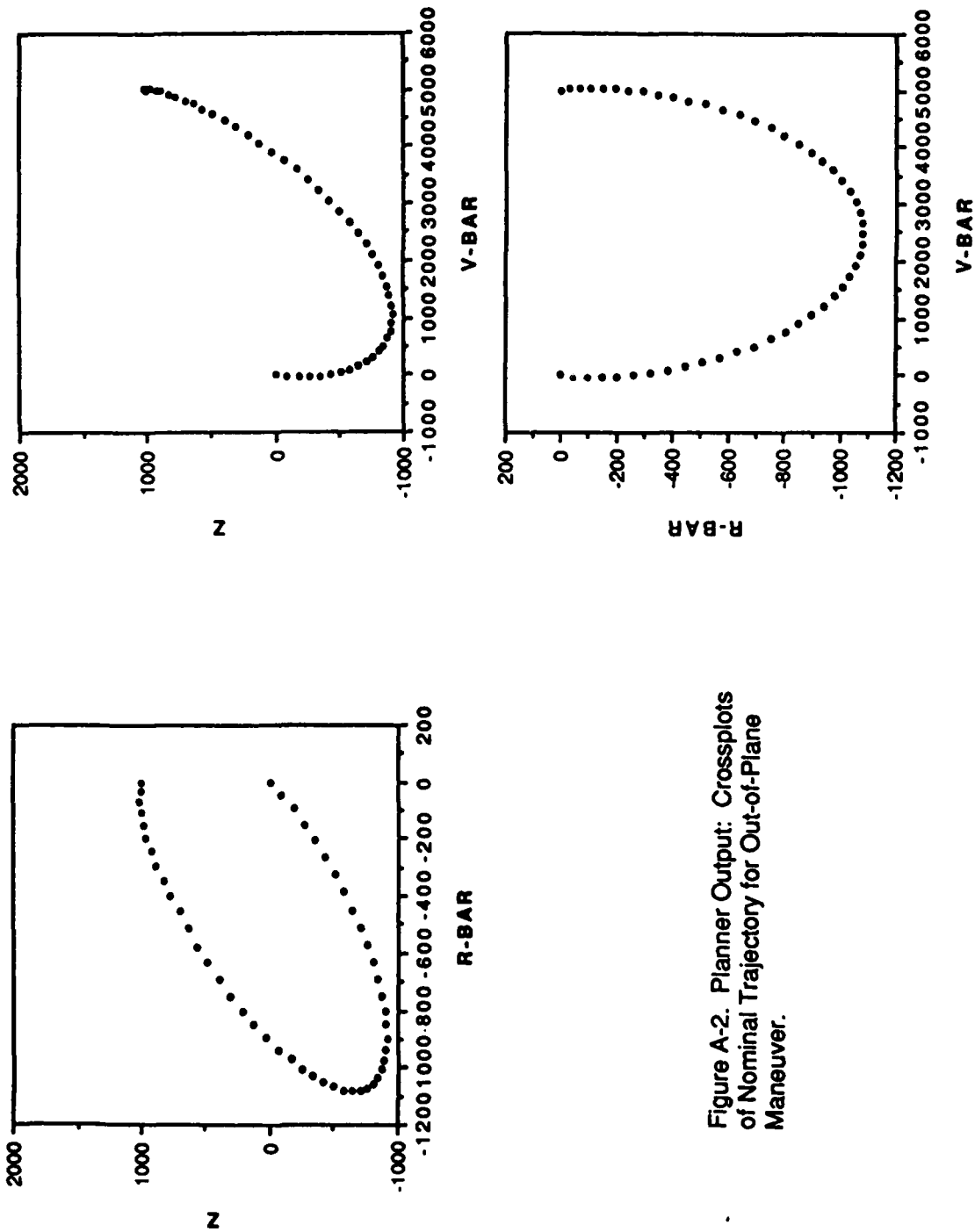


Figure A-2. Planner Output: Crossplots of Nominal Trajectory for Out-of-Plane Maneuver.

Figure A-3. WAYPOINT FILE FOR OBSTACLE AVOIDANCE, OUT-OF-PLANE MANEUVER

X	Y	Z	VX	VY	VZ	TIME	FLAG
5000.00	1000.00	.000000	.258109	.170171	-.262727	.000000	1
5014.34	1007.78	-16.8292	.219034	8.895775E-02	-.298018	60.0000	0
5026.19	1010.67	-35.7333	.175141	7.313238E-03	-.331864	120.000	0
5035.26	1008.65	-56.6207	.126643	-7.436675E-02	-.364101	180.000	0
5041.30	1001.75	-79.3900	7.377578E-02	-.155686	-.394571	240.000	0
5044.03	989.987	-103.931	1.679513E-02	-.236250	-.423128	300.000	0
5043.24	973.422	-130.124	-4.402243E-02	-.315669	-.449633	360.000	0
5038.68	952.137	-157.843	-.108382	-.393556	-.473957	420.000	0
5038.13	950.153	-160.218	-.113895	-.399966	-.475882	425.000	1
5029.27	923.871	-189.437	-.181739	-.475769	-.497711	485.000	0
5016.26	893.107	-219.895	-.252457	-.549264	-.517126	545.000	0
4998.93	858.012	-251.442	-.325706	-.620095	-.534032	605.000	0
4977.13	818.756	-283.927	-.401131	-.687919	-.548349	665.000	0
4950.75	775.529	-317.191	-.478366	-.752407	-.560007	725.000	0
4919.70	728.540	-351.073	-.557036	-.813245	-.568948	785.000	0
4883.89	678.018	-385.410	-.636760	-.870140	-.575130	845.000	0
4880.69	673.656	-388.286	-.647993	-.791162	-.679515	850.000	1
4838.96	624.591	-429.142	-.742854	-.843675	-.681788	910.000	0
4791.54	572.497	-470.035	-.837801	-.892096	-.680754	970.000	0
4738.44	517.626	-510.766	-.932375	-.936191	-.676419	1030.00	0
4679.68	460.245	-551.139	-1.02612	-.975745	-.668804	1090.00	0
4615.33	400.631	-590.958	-1.11857	-1.01057	-.657944	1150.00	0
4545.48	339.075	-630.029	-1.20929	-1.04049	-.643894	1210.00	0
4470.26	275.874	-668.163	-1.29783	-1.06536	-.626720	1270.00	0
4463.75	270.542	-671.292	-1.30927	-.975317	-.720480	1275.00	1
4382.21	211.415	-713.900	-1.40820	-.994812	-.699205	1335.00	0
4294.83	151.261	-755.129	-1.50393	-1.00948	-.674540	1395.00	0
4201.81	90.3746	-794.779	-1.59599	-1.01926	-.646603	1455.00	0
4103.39	29.0495	-832.659	-1.68394	-1.02409	-.615531	1515.00	0
3999.83	-32.4164	-868.583	-1.76735	-1.02395	-.581472	1575.00	0
3891.41	-93.7252	-902.379	-1.84582	-1.01885	-.544594	1635.00	0
3778.44	-154.579	-933.882	-1.91897	-1.00880	-.505075	1695.00	0
<u>3768.83</u>	<u>-159.621</u>	<u>-936.399</u>	<u>-1.92481</u>	<u>-1.00775</u>	<u>-.501668</u>	<u>1700.00</u>	<u>1</u>
3651.30	-219.650	-965.246	-1.99179	-.992405	-.459504	1760.00	0
3529.94	-278.613	-991.495	-2.05273	-.972250	-.415111	1820.00	0
3405.10	-336.225	-1015.02	-2.10735	-.947380	-.368705	1880.00	0
3277.19	-392.207	-1035.70	-2.15538	-.917916	-.320510	1940.00	0
3146.59	-446.286	-1053.45	-2.19658	-.884000	-.270762	2000.00	0
3013.74	-498.201	-1068.17	-2.23076	-.845796	-.219700	2060.00	0
2879.04	-547.700	-1079.79	-2.25775	-.803490	-.167572	2120.00	0
2867.75	-551.708	-1080.62	-2.25967	-.800819	-.151368	2125.00	1
2731.61	-598.380	-1088.11	-2.27707	-.754299	-.9.840056E-02	2185.00	0
2594.65	-642.150	-1092.42	-2.28706	-.704121	-.4.495569E-02	2245.00	0
2457.31	-682.806	-1093.50	-2.28959	-.650527	8.707147E-03	2305.00	0
2320.05	-720.151	-1091.37	-2.28464	-.593779	6.232774E-02	2365.00	0
2183.31	-754.002	-1086.03	-2.27223	-.534150	.115646	2425.00	0
2047.53	-784.197	-1077.51	-2.25244	-.471931	.168404	2485.00	0
1913.16	-810.588	-1065.84	-2.22535	-.407423	.220344	2545.00	0
1902.04	-812.612	-1064.73	-2.22277	-.401954	.224628	2550.00	1
1769.68	-834.739	-1049.72	-2.18792	-.335320	.275400	2610.00	0
1639.63	-852.817	-1031.71	-2.14610	-.267060	.324836	2670.00	0
1512.29	-866.760	-1010.77	-2.09749	-.197504	.372696	2730.00	0

Figure A-3. (Continued)

1388.06 -876.499 -987.018 -2.04234 -.126991 .418749 2790.00 0  
 1267.33 -881.987 -960.562 -1.98091 -5.586122E-02 .462772 2850.00 0  
 1150.47 -883.197 -931.531 -1.91350 1.553911E-02 .504549 2910.00 0  
 1037.82 -880.123 -900.065 -1.84045 8.686411E-02 .543880 2970.00 0  
 1028.64 -879.674 -897.338 -1.83411 9.279357E-02 .547041 2975.00 1  
 920.929 -871.978 -863.408 -1.75533 .163647 .583506 3035.00 0  
 818.095 -860.053 -827.374 -1.67167 .233706 .617140 3095.00 0  
 720.419 -843.956 -789.411 -1.58352 .302632 .647782 3155.00 0  
 628.154 -823.766 -749.703 -1.49132 .370090 .675282 3215.00 0  
 541.532 -799.581 -708.442 -1.39552 .435753 .699507 3275.00 0  
 460.754 -771.518 -665.830 -1.29658 .499303 .720339 3335.00 0  
 385.995 -739.713 -622.071 -1.19498 .560431 .737678 3395.00 0  
 380.041 -736.899 -618.380 -1.18641 .565406 .738962 3400.00 1  
 311.965 -701.215 -573.620 -1.08249 .623577 .752422 3460.00 0  
 250.176 -662.130 -528.162 -0.976938 .678723 .762232 3520.00 0  
 194.755 -619.834 -482.226 -0.870281 .730578 .768346 3580.00 0  
 145.754 -574.531 -436.035 -0.763032 .778889 .770733 3640.00 0  
 103.192 -526.443 -389.813 -0.655710 .823422 .769382 3700.00 0  
 67.0595 -475.801 -343.784 -0.548837 .863962 .764299 3760.00 0  
 37.3126 -422.851 -298.171 -0.442931 .900312 .755510 3820.00 0  
 35.1199 -418.342 -294.396 -0.434165 .903147 .754611 3825.00 1  
 12.2073 -363.183 -249.484 -0.329885 .934759 .741855 3885.00 0  
 -4.50690 -306.262 -205.445 -0.227634 .961837 .725500 3945.00 0  
 -15.1593 -247.856 -162.494 -0.127907 .984251 .705627 4005.00 0  
 -19.9159 -188.247 -120.839 -0.3118826E-02 1.00189 .682332 4065.00 0  
 -18.9714 -127.726 -80.6806 6.205315E-02 1.01467 .655727 4125.00 0  
 -12.5481 -66.5852 -42.2150 .151365 1.02253 .625942 4185.00 0  
 -.894876 -5.12140 -5.62828 .236314 1.02543 .593121 4245.00 0  
 .000000 .000000 .000000 .000000 .000000 .000000 4250.00 1

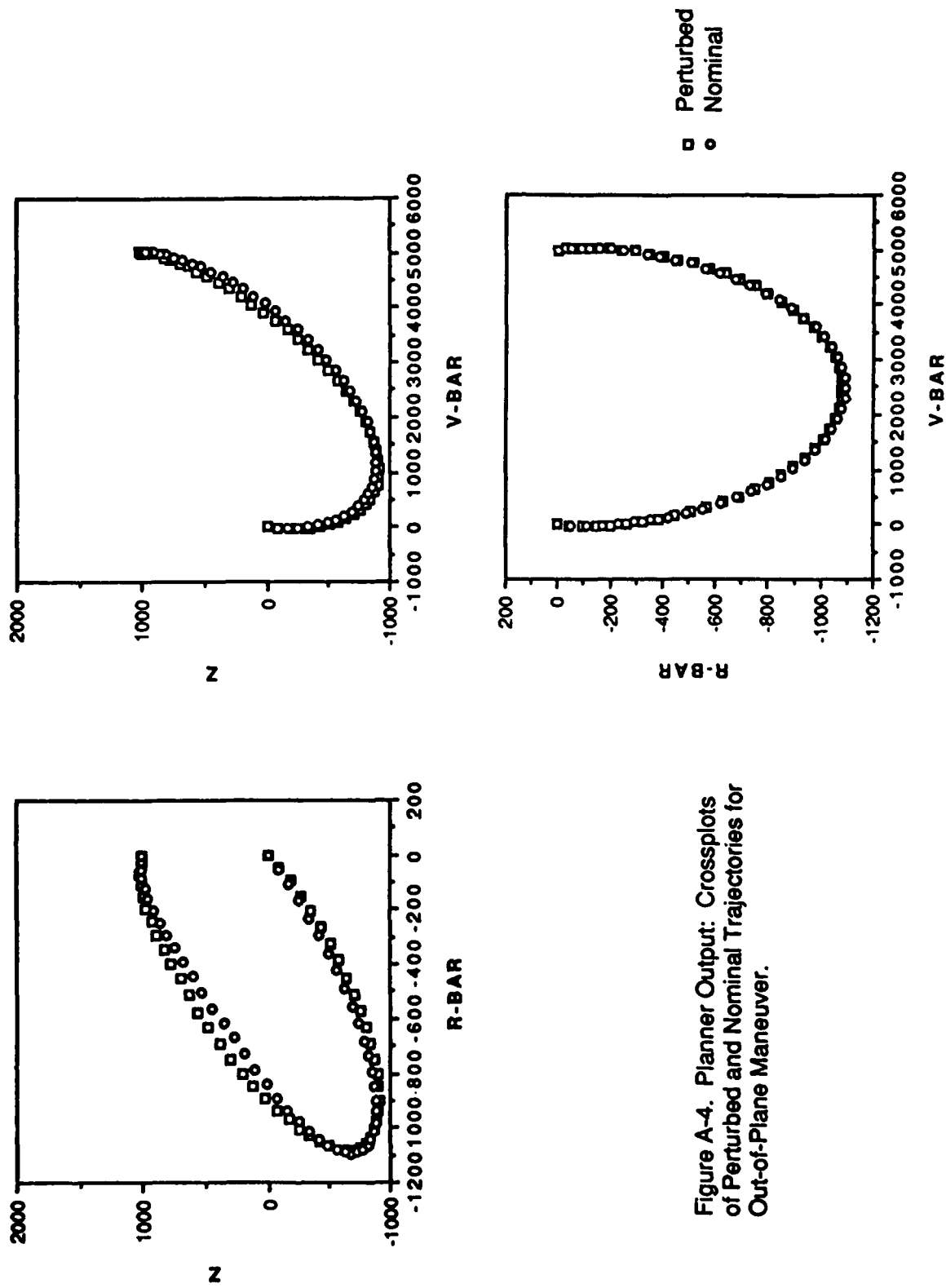


Figure A-4. Planner Output: Crossplots of Perturbed and Nominal Trajectories for Out-of-Plane Maneuver.



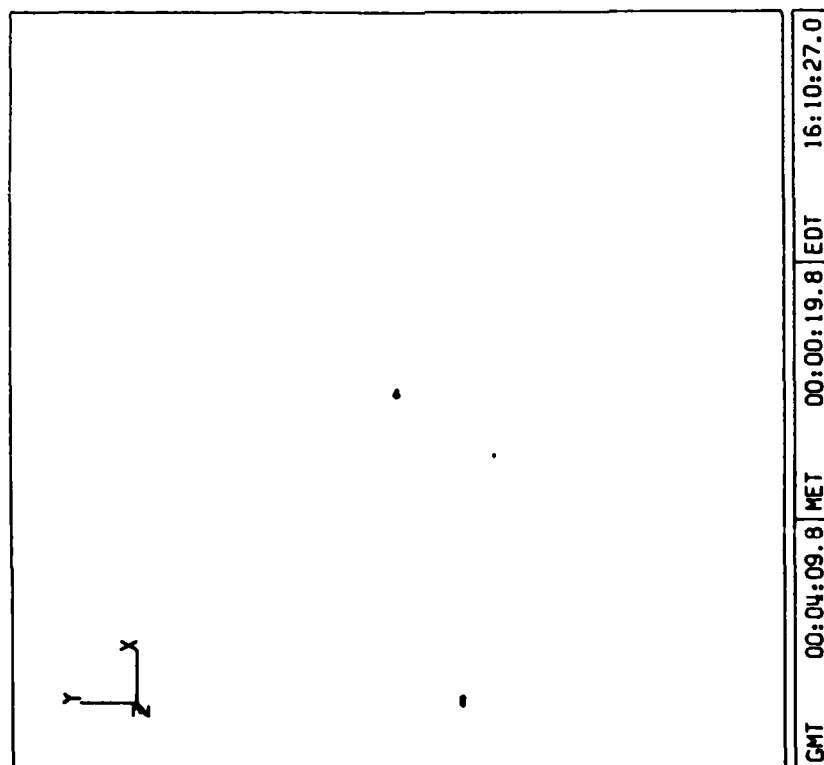
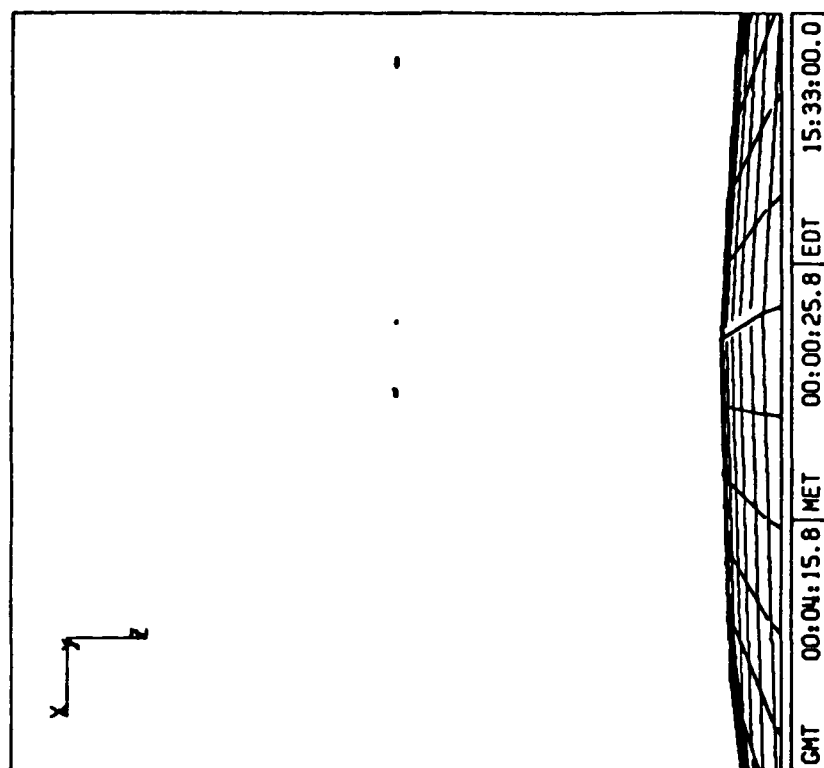


Figure A-5. Sequential Frames of Out-of-Plane Maneuver (1 of 8)

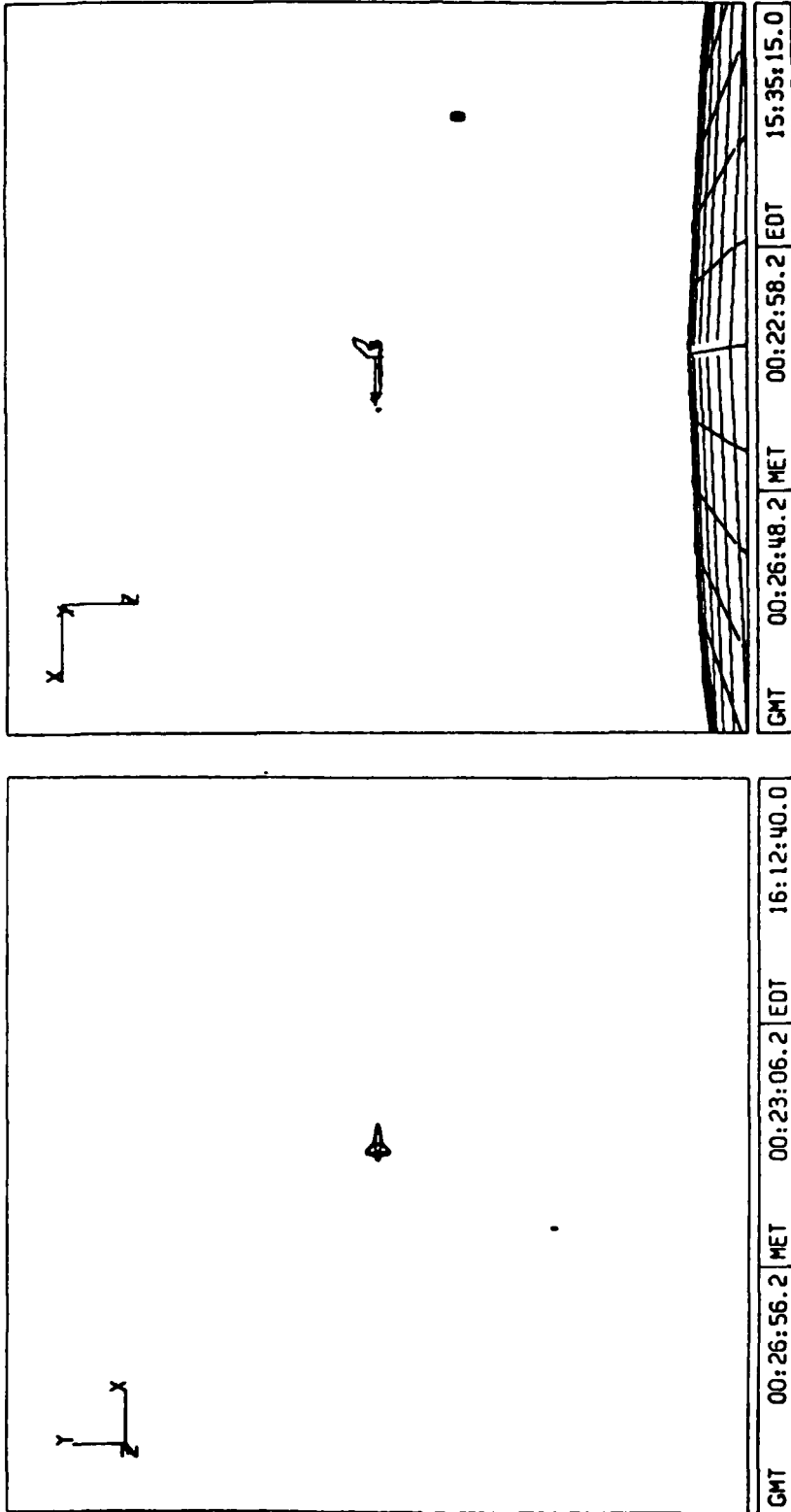


Figure A-5. Sequential Frames of Out-of-Plane Maneuver (2 of 8)

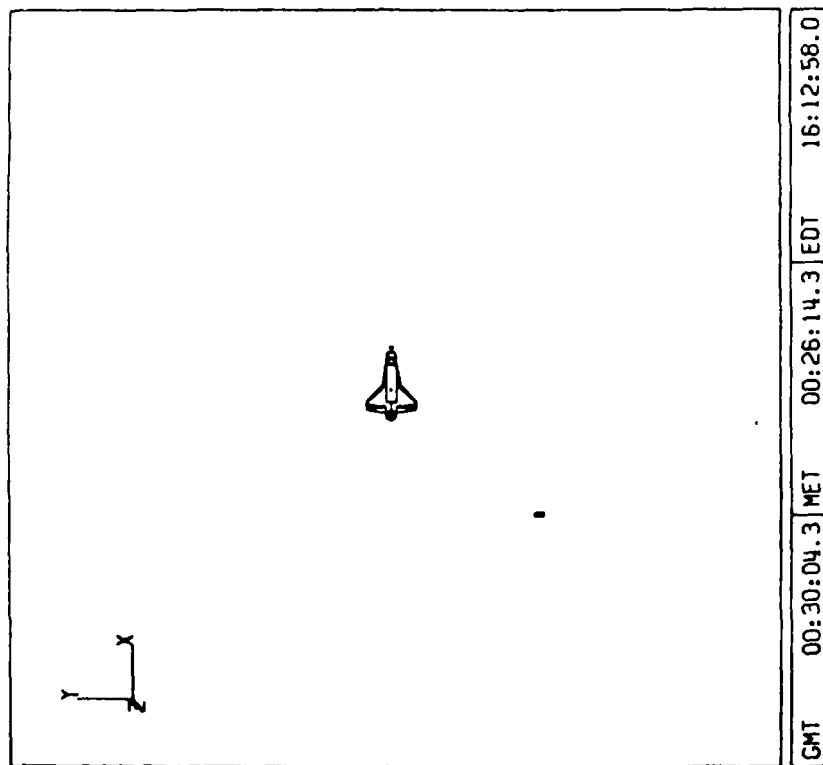
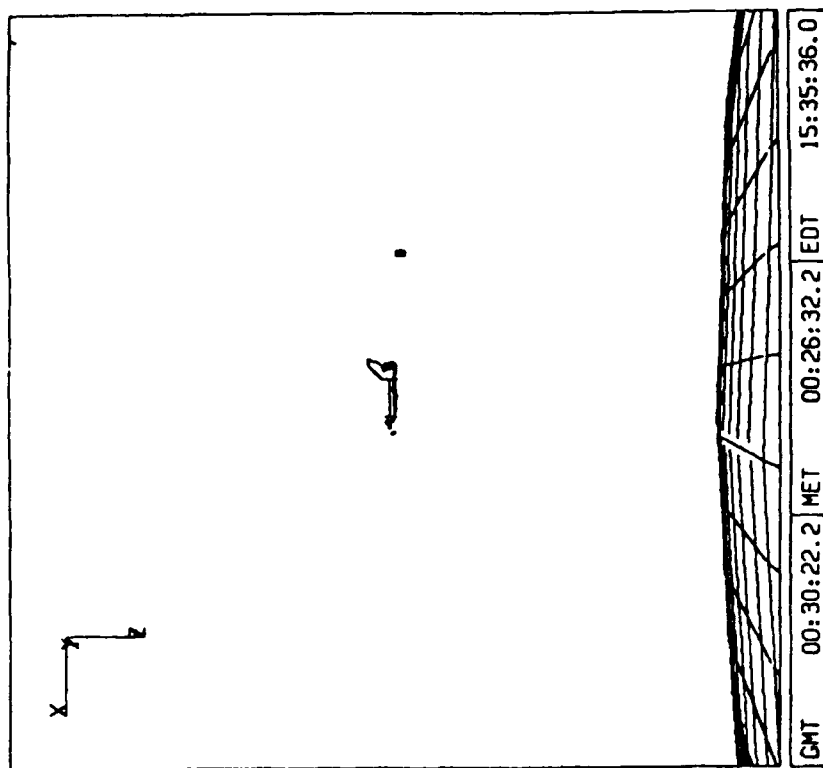


Figure A-5. Sequential Frames of Out-of-Plane Maneuver (3 of 8)

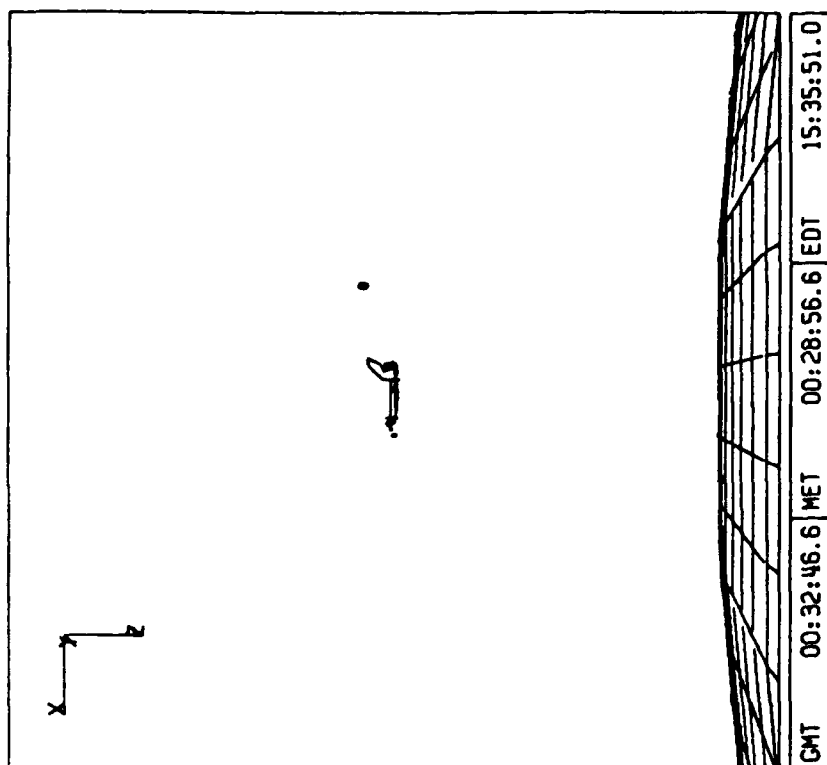
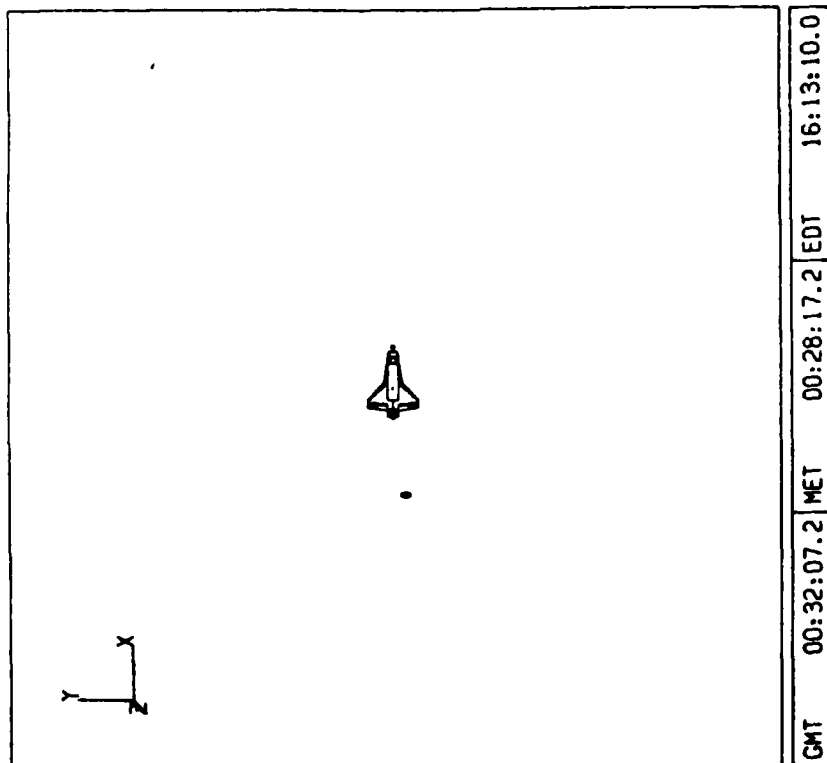


Figure A-5. Sequential Frames of Out-of-Plane Maneuver (4 of 8)

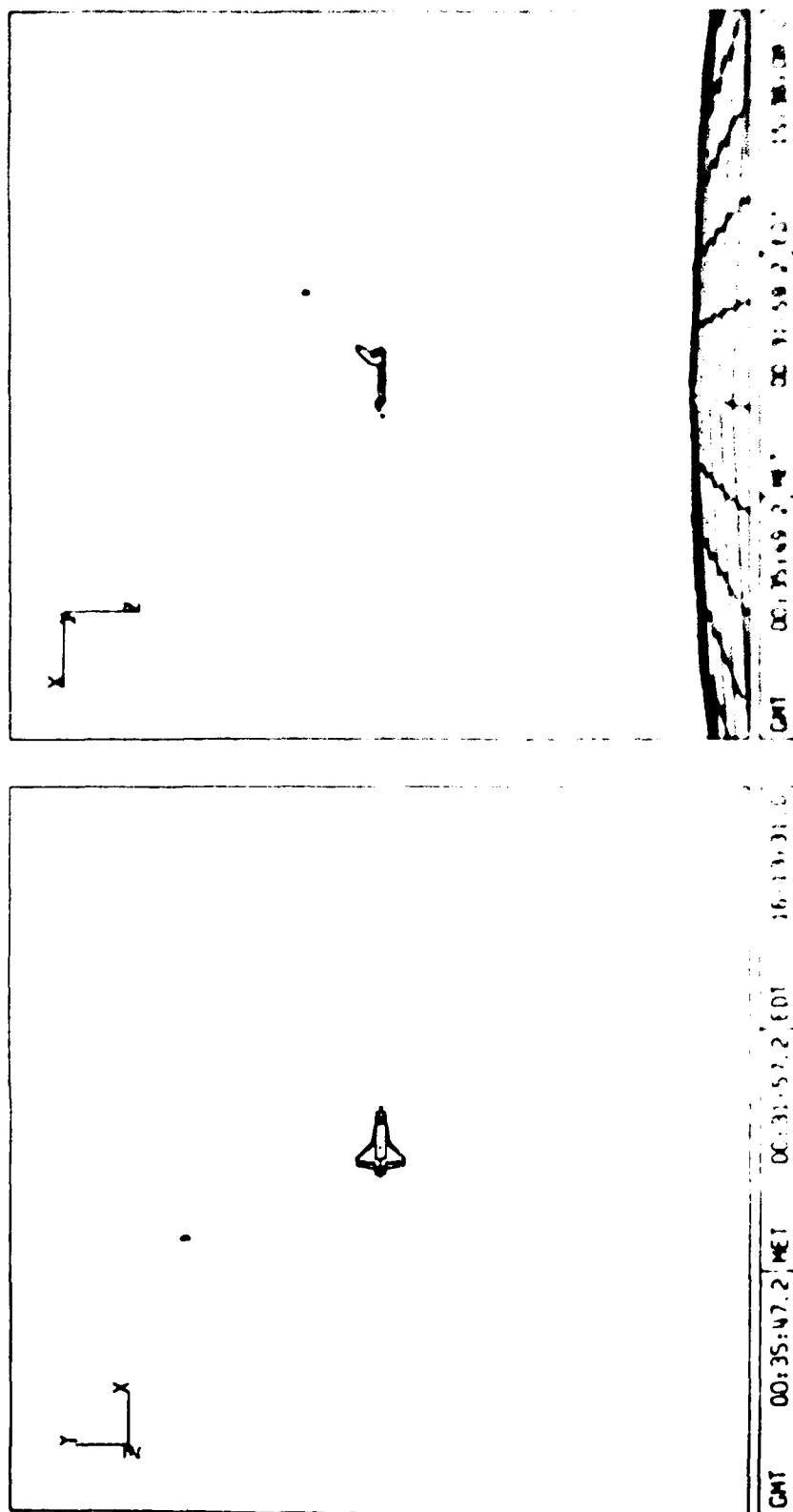


Figure A-5. Sequential Frames of Out-of Plane Maneuver (5 of 8)

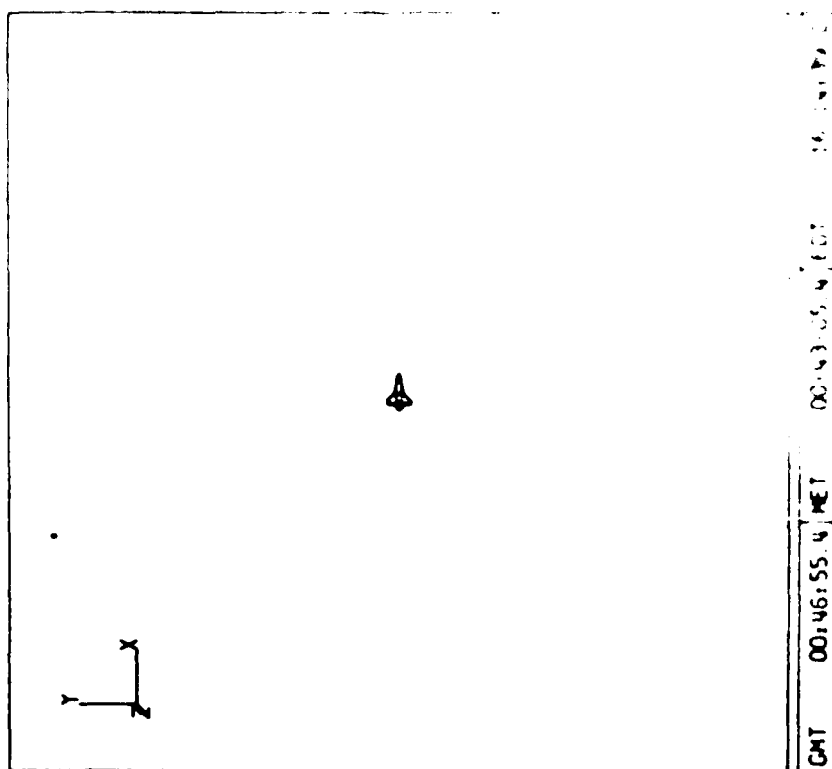
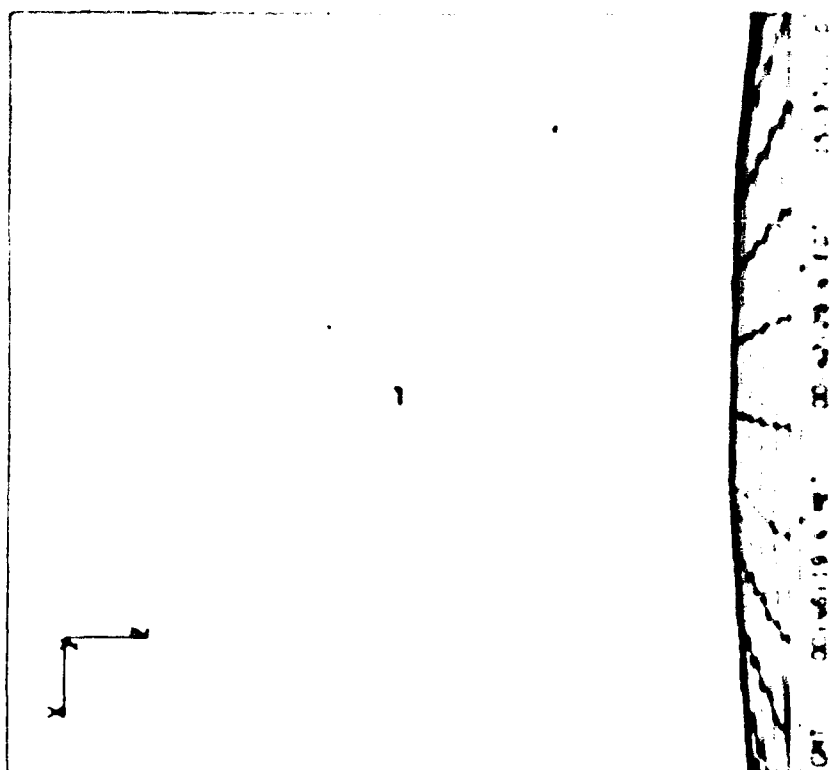
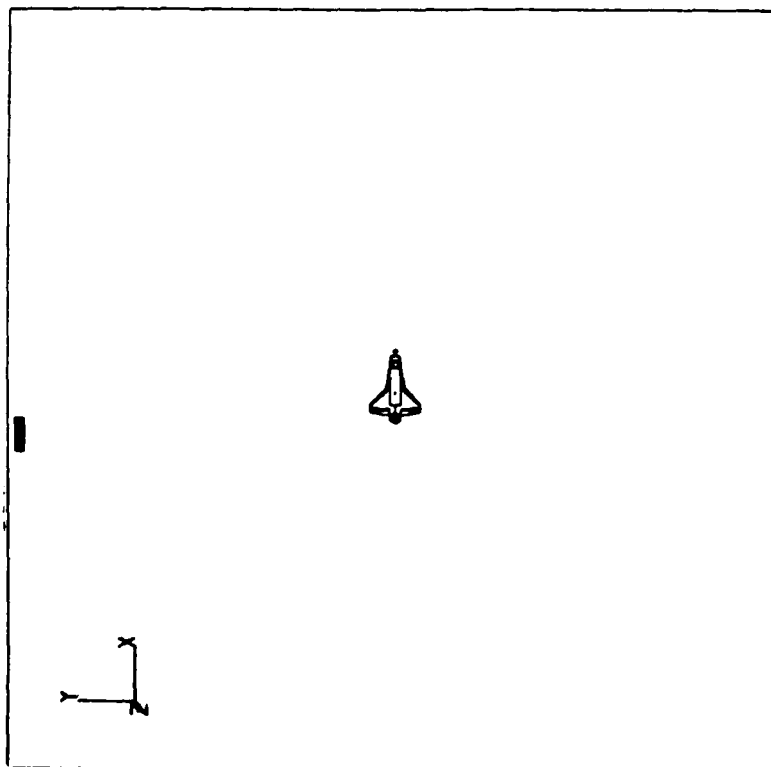
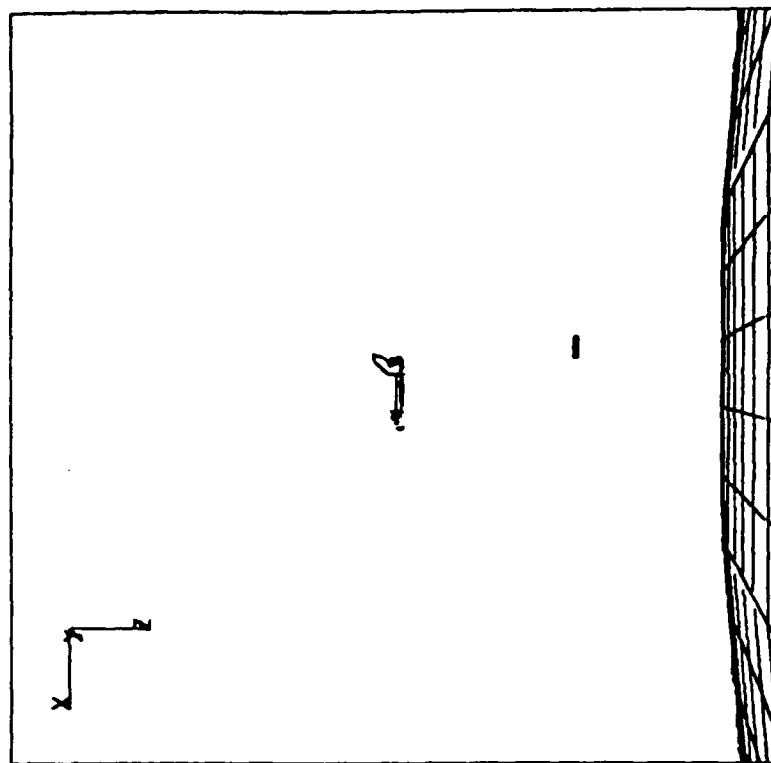


Figure A-5. Sequential Frames of Out of Plane Maneuver (6 of 8)



GMT	01:06:57.2	MET	01:03:07.2	EDT	16:34:52.0
-----	------------	-----	------------	-----	------------



GMT	01:06:01.1	MET	01:02:11.1	EDT	15:39:09.0
-----	------------	-----	------------	-----	------------

Figure A-5. Sequential Frames of Out-of-Plane Maneuver (7 of 8)

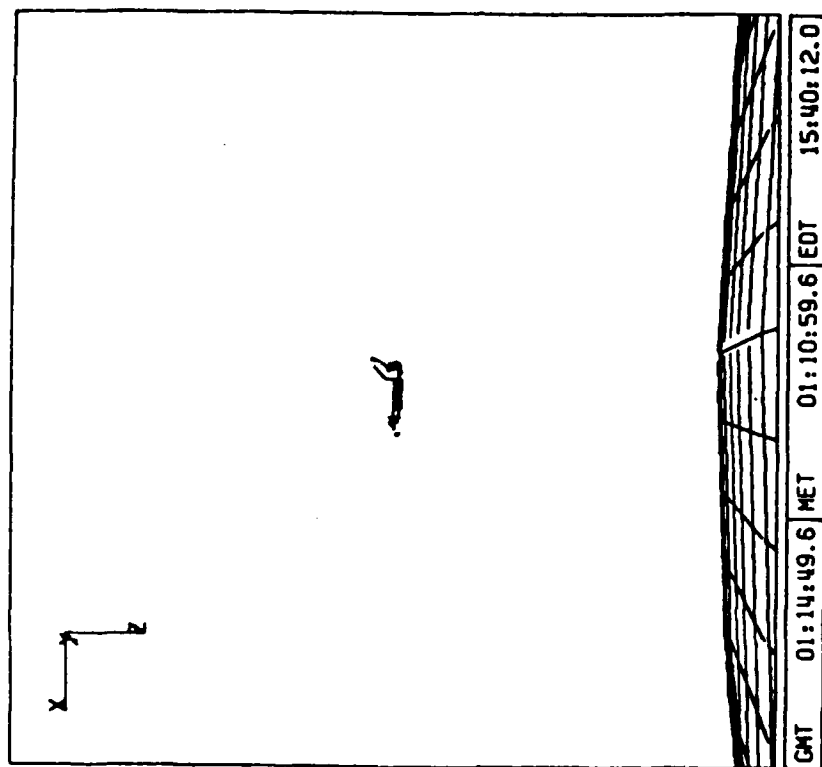
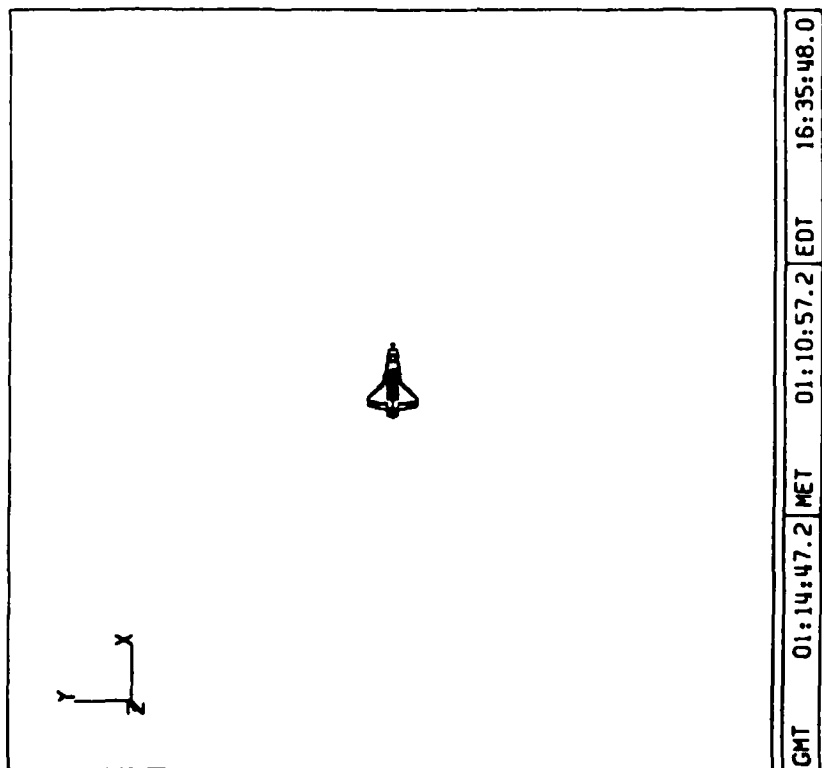


Figure A-5. Sequential Frames of Out-of-Plane Maneuver (8 of 8)



## REFERENCES

1. Vaughan, R.M. and E.V. Bergmann, "Manually Augmented Proximity Operations and Docking Control," AIAA paper 85-1941 presented at the AIAA 1985 Guidance, Navigation and Control Conference.
2. Fehse, W., "Technology Development by the European Space Agency for Autonomous Spacecraft Rendezvous and Docking", presented in the MIT / Draper Seminar Series on Dynamics, Guidance and Control, October 31, 1988.
3. Vaughan, R.M. "Collision Detection For Spacecraft Proximity Operations," PhD Thesis, Department of Aeronautics and Astronautics, MIT, May 1987.
4. Bergman, E.V. et al., "An Advanced Spacecraft Autopilot Concept," Journal of Guidance and Control, Vol. 2, No. 3, May-June 1979.
5. Deutsch, O.L., J.V. Harrison, and M.B. Adams, "Heuristically Guided Planning for Mission Control/Decision Support," AIAA 1985 Guidance, Navigation and Control Conference.
6. Deutsch, O.L., M. Desai, and L.A. McGee, "Far-Field Mission Planning for Nap-Of-The-Earth Flight," presented at 1987 American Helicopter Society Conference on Rotorcraft Flight Controls and Avionics.
7. Kaplan, Marshall H. Modern Spacecraft Dynamics and Control. John Wiley & Sons, Inc, 1976.
8. Battin, Richard H. An Introduction to the Mathematics and Methods of Astrodynamics. AIAA, Inc., 1987.
9. Clohessy, W.H. and R.S. Wiltshire, "Terminal Guidance System for Satellite Rendezvous," Journal of the Aerospace Science, Sept. 1960.
10. Redding, D.C., B.A. Persson, and E.V. Bergmann, "Combined Solution of Spacecraft Rotational and Translational Maneuvers", presented at the AIAA 1986 Guidance, Navigation and Control Conference.

11. Pearl, J. Heuristics: Intelligent Search Strategies for Computer Problem Solving. Addison-Wesley Publishing Co., 1984.
12. Bergmann, E. V., and P. Weiler, "Accommodation of Practical Constraints by a Linear Programming Jet Select," AIAA 1983 Guidance and Control Conference.
13. Kirk, Donald E. Optimal Control Theory. Prentice-Hall Inc., 1970.
14. Glandorf, D.R., "An Innovative Approach to the Solution of a Family of Linear Programming Problems," Lockheed Engineering and Management Services Co., Memo APO 147, December 1987.
15. Leonard, Carolina L. "Formationkeeping of Spacecraft Via Differential Drag," SM Thesis, Department of Aeronautics and Astronautics, MIT, July 1986.
16. Polutchko, Robert J. "Cooperative Control of Two Active Spacecraft During Proximity Operations," SM Thesis, Department of Aeronautics and Astronautics, MIT, August 1989.